# Functions and structure of a packet radio station*

*by* J. BURCHFIEL, R. TOMLINSON and M. BEELER

*Bolt Beranek and Newman*
Cambridge, Massachusetts

## INTRODUCTION

A packet radio network is a digital broadcast channel, fixed and mobile digital terminals which are sources and sinks of information, stations which provide centralized routing control and interconnections to other networks, and repeaters which provide area coverage for mobile terminals by performing a store-and-forward function on the radio broadcast channel. An experimental testbed for these concepts is now under construction at the Stanford Research Institute, Palo Alto, California to provide experimental validation of theoretical models and simulation predictions.[1-6]

Objectives of this technology exploration include low-cost use of a broadcast band in digital burst mode to support digital computer and terminal communication, demonstration of coexistence with existing broadcast applications, and secure mobile communications through encryption techniques.

The Packet Radio Network (PRN) concept was initially explored in the ALOHA Project at the University of Hawaii.[2] Such a network is different from point-to-point packet switched networks in a number of significant areas:

1. The use of mobile terminals requires facilities for tracking and handoff of the terminals from repeater to repeater and station to station.
2. The use of a common broadcast channel results in collisions due to simultaneous packet transmissions: the communications paths require extensive error checking and correction to handle the resulting high error rate.
3. The repeater is required to operate unattended in relatively remote areas for long periods of time. It must therefore be a simple, low-powered component. Accordingly, it is not designed to support complex dynamic routing algorithms. Instead, such functions must be provided in the stations, giving some measure of centralized control in the Packet Radio Network.

On the other hand, a PRN has many features and functions in common with a point-to-point packet switched network:

1. The PRN must provide routing which adapts dynamically to component failures.
2. The PRN must support interprocess connections which have flow control and error control.
3. The PRN must have centralized monitoring, debugging, and statistics collection tools to provide maintenance and performance evaluation capabilities.

The dynamic packet routing capability (packet store-and-forward) is programmed in the repeaters, with the stations providing initialization and centralized control of parameters for terminal tracking. The programmable capability of the repeater is provided in its IMP-16 microprocessor.

Reliable data transmission between PRN data sources and sinks is required in spite of errors and transmission 'collisions' on the broadcast channel. This is achieved by defining a logical entity called a 'connection' between the source process and destination process, and performing end-to-end error detection and correction over this noisy channel. The connection is thus a reliable (error corrected) data transmission path.

The programs for providing interprocess connections within the PRN must be programmed into each terminal (data connection), each repeater (control connection), and each station (data and control connections).

In addition to the communications support, the terminal must also have a terminal handler program which manages terminal input and output buffers and performs translation of format effector characters as needed.

The programs which provide centralized monitoring, debugging, and statistics collection are located in the station, with a small (slave) routine in each repeater. These functions are shown in Figure 1.

The set of functions which appear in common in a station, repeater, and terminal are identified in Figure 1 as a Packet Radio Unit, or PRU, which has been implemented as a standard piece of hardware and software by Collins Radio. It will serve standalone as a repeater; addition of the station interface hardware and software option converts it to a station; addition of the terminal interface

* This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract No. DAHC15-71-C-0088.

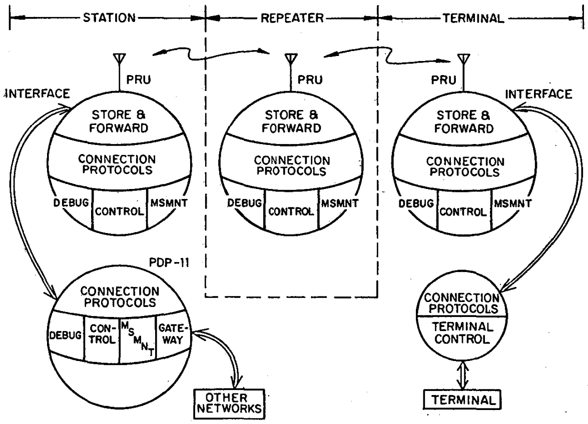


Figure 1—Functions of a packet radio station, repeater and terminal

hardware and software option converts it to a terminal. The additional functions shown for terminal may either be implemented in a separate microprocessor or provided in a separate memory partition within the terminal's PRU, timesharing its microprocessor for economy.

Our prototype station has the additional station functions implemented in a Digital Equipment Corp. PDP-11, which is also interfaced to the ARPANET.

An implementation of the station described in this paper will provide control functions for an experimental test of packet radio concepts in a testbed packet radio network in the Palo Alto, California area during 1975.

## CONNECTIONS IN THE PACKET RADIO NETWORK

One of the basic facilities required in the PRN is support of interprocess connections which provide reliable delivery of data from a PRN source to a PRN destination. Such connections require flow control to prevent a source from overloading the network and causing serious congestion. They also require error control because of message interference in the shared broadcast communication channel. The error control mechanisms selected are:

1. A sequence number in each packet to permit detection of missing or duplicate packets.
2. An end-to-end positive acknowledgment for packets which arrive successfully.
3. A source timeout which causes periodic retransmissions of unacknowledged packets.

Since most expected uses of the PRN will require bi-directional communication, the PRN connection is bi-directional, with flow control and error control information for data transmission in each direction piggybacked onto the data flow in the opposite direction. This arrangement is depicted in Figure 2.

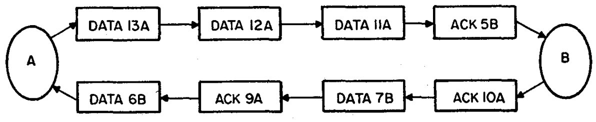
Certain applications of packet radio will not require reliable delivery of data (e.g., real time seismic or speech data), and such devices will not be required to support the connection protocol described above.

Figure 2 shows that multiple packets of data can be in transit in both directions at one time. The acknowledgments are cumulative, i.e., an acknowledgment carrying serial number 9 acknowledges successful receipt of all packets up to and including packet number 9. It is therefore unnecessary to send out an acknowledgment for each data packet received; one 'ack' can cover a number of data packets. Further, the 'ack' carries such a small amount of information, that it can be inserted into the header of a data packet travelling in the same direction, resulting in very low overhead for this acknowledgment mechanism. However, timely acknowledgment is required to prevent source timeout and retransmission, so if there is no data traffic which can be used in this piggyback fashion for 100 milliseconds or so, a separate 'ack' packet must be generated and sent.

The status information which must be maintained at each end of the connection is minimal: four sequence numbers. The values of the sequence numbers at the two ends differ by the traffic currently 'in the pipeline'. Flow control is established by the convention that the sender can only send up to N packets ahead of the last packet which was acknowledged. Equivalently, a source may not have more than N packets 'in the pipeline' at one time. To keep the repeater code as simple as possible, N should be equal to one (packet-at-a-time) for repeater control connections. This protocol is a simplified subset of the protocol developed by Cerf and Kahn[7] for internetwork communications.

The protocols of the Packet Radio Network are layered, or hierarchical. The program which deals with control information at level M passes control and data for all levels greater than M as transparent data. Conversely, the program which deals with control at level M does not see control information at levels less than M; it is inserted by lower level programs on transmission, and stripped off by lower level programs on reception.

Figure 3 shows this layering explicitly: the connection protocol described above is the level 2 protocol, based on the level 1 routing protocol which controls the PRN store-and-forward routing for the packet. The routing protocol is itself based on a level 0 'Radio Hop' protocol which provides broadcast synchronization and error detection for transmission of the packet from one PRU to the next.



| A STATUS: | | B STATUS: | |
|---|---|---|---|
| SENT | RECEIVED | SENT | RECEIVED |
| DATA 13A | ACK 8A | DATA 7B | ACK 4B |
| ACK 5B | DATA 5B | ACK 10A | DATA 10A |

Figure 2—Full duplex, point-to-point connection

The sequence numbers shown were discussed previously. The 'sequence control' field is used to synchronize sequence numbers when a connection is established, and to signal termination of the connection. The 'function fields' provides an address: within a PRU, it selects the control process, the debugging process, or the measurement process.

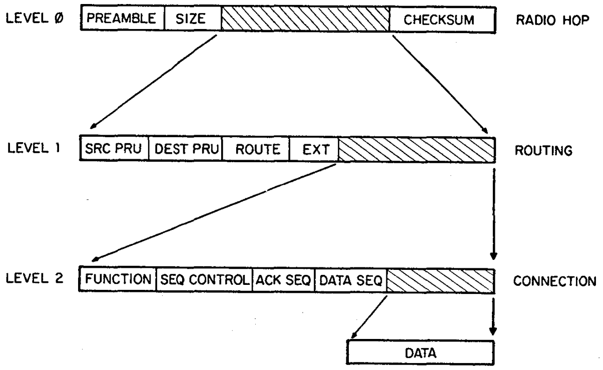


Figure 3—Protocols for support of a PRN connection

## STATION CONTROL FUNCTIONS

The control functions performed by a station include initialization of the PRN, dynamic routing changes, and multi-station coordination. Initialization of the PRN includes the following steps:

1. Measurement of RF propagation connectivity between all stations and repeaters. This measurement data is used to construct the *connectivity matrix*: this is a matrix of binary values which indicate the radio units which are capable of direct communication with each other.
2. Configuring the PRN by loading each repeater with routing parameters which control the packet store-and-forward program. These parameters specify forwarding of packets [in the direction of minimum distance] to the next repeater within "earshot".
3. Establishing control, debugging, and measurement connections from the station to each repeater that it controls. These connections remain open to perform the indicated functions as long as the station and repeater continue to function normally.

The initialization algorithm must be able to bring up the network from a cold start, with no information in any station. It must permit any element to enter the net 'at any time on its own initiative, e.g., when a terminal is powered up the algorithm must connect it into the net. This algorithm should also provide a continuous monitor on connectedness to identify component failures. Finally, it should be quick, accurate and use minimum traffic.

The proposed procedure to meet these requirements is as follows:

1. Repeaters have two states: labelled (containing routing parameters) and unlabelled.
2. Each PRU (station, repeater, terminal) has a unique, hardwired I.D.
3. All repeaters send "search" packets at random times with some average rate dependent on whether or not the repeater is labelled.
4. A search packet states whether or not the repeater is labelled, gives its repeater I.D., and the I.D. of the station which labelled it.
5. All labelled repeaters that hear a search packet on the first hop forward it via their established route to every station that has labelled the receiving repeater. All unlabelled repeaters ignore search packets.
6. The station uses search packets arriving to generate the connectivity matrix, and relabel (or initially label) the net.
7. On initial entry by a station into an unlabelled net:
   a. The station listens for at least one search packet.
   b. If it hears none it should try to stimulate receivers in earshot.
   c. The first search packet heard is labelled as soon as heard.
   d. The next search packet may come through the labelled PRU or directly or both. The station labels the new PRU and notes whether the first PRU could hear it, and starts the connectivity matrix.
   e. The station continues to label or relabel as repeaters report in.
8. If an ability to trigger search packets is available, the process can be speeded up. This should allow stations entering a labelled net (one where another station is operating) to do so more rapidly. Repeaters should also have ability to trigger local repeaters, or to get local repeaters to bounce back search packets. This will occur in the above process for all labelled local repeaters, but requires special consideration for unlabelled PRU's.

Once the station has labelled all PRU's and established connections to them, the information for maintaining these connections is entered into the station's *connection table*. This contains the status information described above for handling the connection protocol. As terminals come "on-line" within the PRN, each terminal is also given a connection to its controlling station, and this information is added to the station's connection table.

Dynamic routing changes are performed locally within the PRN by permitting a repeater to specify an alternate address for the next hop after some number of unsuccessful attempts to forward the packet along its specified route. This capability provides a localized terminal tracking facility which will hand off a mobile terminal from one repeater to another.

A less dynamic but more global routing change is required when a repeater or station fails: persistent alternate routing of packets signals this failure, and the station detecting it must reconfigure the PRN to route all traffic around the failed element. The station performs this reconfiguration by updating the connectivity matrix to indicate that the failed element is incommunicado; by recomputing minimum distance routes to the elements which are still active; and finally, by updating repeater routing parameters to route packets along these new routes.

The third station control function is multi-station coordination. This will be supported by PRN connections between stations and a station-station protocol. In some cases, when both stations are connected to a point-to-point network, it may be more efficient to use this network for station coordination rather than the PRN. This coordination is required in two cases:

1. Hand off of a mobile terminal as it passes out of the area controlled by one station and into the area controlled by another station.
2. An alternate station assuming all communications responsibilities for a station which fails.

These applications will require a distributed data base stored redundantly at a number of stations. The data will give status information on each active connection, and will be updated often enough that little information will be lost in the event of a station failure. Some of the PRN terminals may be accessing host resources in the ARPANET: an ARPANET reconnection protocol will be defined to permit switching the ARPANET portion of the terminal connection from the failed station to the backup station. These complexities are not being addressed in the initial experimental implementation, but they will be vital reliability measures in an operational packet radio network. Many of these issues of reliability achieved through maintenance of redundant distributed data bases have been explored in the RSEXEC (Resource Sharing Executive) distributed computation testbed.[10]

A second phase of the experimental program being conducted at the Packet Radio testbed network at Palo Alto will demonstrate the concepts of redirecting traffic from a failed station to an operational station, and the backup of resource allocation information.

## STATION DEBUGGING AND STATISTICS FUNCTIONS

A level-3 debugging protocol has been defined which supports debugging of remote PRU's from terminals attached to a central station. The debugging functions include examining and depositing words into the PRU memory, and setting "mousetraps" which send an error code to the controlling station when some anomalous condition occurs, e.g., hardware failure. These functions make it possible to do centralized software maintenance of remote, unattended repeaters. The maintenance terminal of the station will normally be attended by an operator or system programmer.

A similar mechanism permits centralized collection of traffic statistics, both through examination of counters in PRU memory and through centralized reception of special status conditions such as "trace packets" moving through the network. Again, it was essential to centralize this function for remote, unattended repeaters.

Once the station has collected a set of traffic statistics, it will normally forward these measurements to a service host for detailed statistical analysis, logging and plotting.

## STATION SUPPORT OF NONENCRYPTED TERMINALS

Some PRN applications will require secure or private communications, i.e., end-to-end encryption between a terminal and the service host which it is accessing. Assuming that the station is merely an intermediate node in the path from terminal to host, the station must be completely transparent to the scrambled data. (Any modification of the data could make it impossible to unscramble.)

On the other hand, some PRN applications do not require encrypted terminals: for these applications, the station can take over part of the terminal service function and simplify the terminals. One example of this is the TELNET RCTE option. [Reference 9] TELNET is the protocol which translates a variety of different physical terminals into a single standard logical terminal called a network virtual terminal: this conversion may involve both code conversion and interpretation of format effector characters (tab, carriage return, etc.). This concept provides and extremely valuable simplification because previously N*M conversion routines were required to interface N terminals to M systems. Using the network virtual terminal concept, only N+M conversion routines are required.

The RCTE option of TELNET is Remote Controlled Transmission and Echoing. It permits character echoing for a full duplex connection to be performed local to the terminal (eliminating annoying echoing delays) under control of the remote service host, which may, for example, suppress echoing of typed-in passwords.

The TELNET RCTE protocol option may be too complex to incorporate into the simplest unencrypted PRN terminal. In this case, the station can handle the protocol conversation with the remote server on the terminal's behalf. Of course, the station cannot perform this service for encrypted terminals.

The TELNET process in the station may also be used to specify and set up connections to remote server hosts via a gateway connection.

## INTERNETWORKING APPLICATIONS

So far, we have focussed on the attachment of terminals to a PRN. It is also reasonable to attach service host computers to a PRN, particularly when the network may be deployed in the absence of other communications networks, (e.g., for fleet communications). Such a host attaches to the PRN as a multiplexed set of the standard PRN connections described earlier.

When some other network is present, it is important to provide connections between the terminals and hosts of the PRN and the terminals and hosts of the other network. This is being done for the ARPANET in two ways:

1. For communication with ARPANET hosts which support a protocol congruent with the PRN connection protocol (the Cerf-Kahn protocol mentioned previously qualifies here), the station functions as an extremely simple gateway: arriving packets are simply forwarded into the other network after their header format is converted to that of the destination network. In this case, the station does not detect missing or duplicate packets, and does not reorder packets which arrive out of order; it is merely a packet reformatting and readdressing service.
2. The second approach will be conversion between the host-host protocols of the two networks. In particular, one connection will be established from a PRN device to a station using the PRN connection protocol described in an earlier section. Another full-duplex connection will be established from the station to an ARPANET host using the current ARPANET host-host protocol. Data arriving from either of these connections will be forwarded through the other connection.

The first approach has four obvious advantages beyond its simplicity: first, error control is truly end-to-end instead of two path (PRN device-to-station, station-to-ARPANET device). Failure of the station will not cause data loss with this approach though it would with the two-path approach.

The second advantage of this approach is dynamic rerouting after a station failure. Since the station is merely doing packet readdressing, any other station attached to the PRN and the ARPANET can do this function equally well; no connection status information is kept in the station. Recovery and continued operation of a connection after the station fails merely requires redirecting the packet traffic to another gateway station.

The third advantage of the "simple gateway" approach is that the gateway does not introduce additional delay into the end-to-end path by forcing reassembly and reordering of packets at an intermediate location (the station).

The fourth advantage of the "simple gateway" approach is that it supports end-to-end encryption of all data except the address headers. This can provide security against data disclosure, but no security against traffic pattern and volume analysis. The second gateway approach would require a secure station to decrypt and re-encrypt data flowing through it.

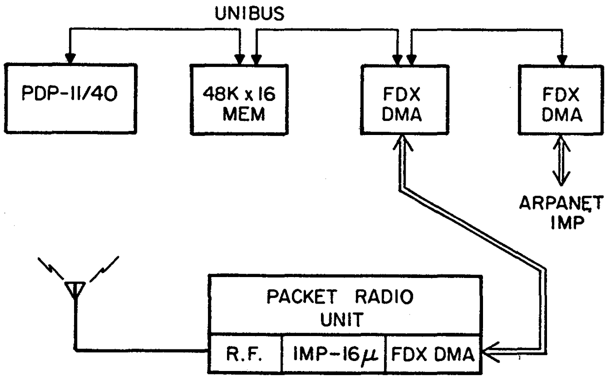


Figure 4—Station hardware

On the other hand, the second approach enjoys two advantages: first, there is independent flow control on the two connections, so 'acks' for PRN packets are returned from the station (about 200 msec) rather than from the remote host (about 1 sec). This faster turn-around of acknowledgments permits shorter source timeouts and more rapid recovery from lost packets.

The second advantage of this approach is that it permits the utilization of completely different protocols which are currently operational within the two networks, as long as the two protocols are functionally equivalent and a mapping exists between the two. This applies not only to host-host protocol, but also to higher level protocols such as TELNET, file transfer, and remote job entry. All that is required is a conversion program for the high-level protocol of interest, interposed between the PRN connection and the ARPANET connection.

The experiments planned for the packet radio testbed system will explore the tradeoffs and performance advantages of these two different gateway concepts in detail.

## STATION STRUCTURE

Figure 4 shows the hardware organization of our prototype station: it is a PDP-11 processor interfaced to a packet radio unit. In the initial tests, it will also be connected as a gateway to the ARPANET. The hardware interface between the PDP-11 and the PRU consists of a pair of memory channels on each end, permitting full duplex DMA packet transfers.

The software organization inside the PDP-11 is shown in Figure 5. We selected the ELF operating system* as the basic environment. This is a time-sharing operating system written in PDP-11 assembly language. Two modifications were required to ELF to support the packet radio application: first, the PRU was added as a bidirectional ELF device. This required addition of an interrupt service routine for the PRU interface hardware, and send and

* Developed by the Speech Communication Research Lab., Santa Barbara, California.

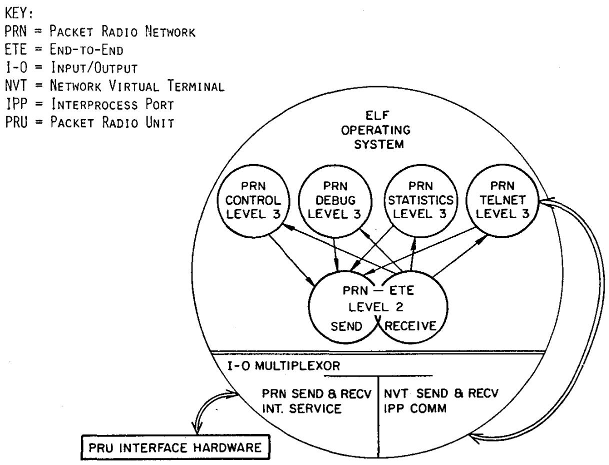


Figure 5—Station software

receive subroutines within the ELF I-O multiplexor module. These routines were coded in assembly language and integrated into the ELF operating system. The second change required to ELF was the 'Network Virtual Terminal' support package which permits PRN terminals to appear identical to local ELF terminals (attached to the PDP-11) by use of a connection to the PRN terminal, a protocol-handling PRN TELNET process, and ELF's interprocess communication facility. These routines were also coded in PDP-11 assembly language and integrated into ELF. The use of internetwork protocol[7] will permit PRN terminals to access services of hosts on other networks.

It was possible to do the remaining packet radio software as independent user applications processes which execute under ELF. Accordingly, they were programmed in BCPL (Basic Compatible Programming Language) to obtain all the clarity, debugging, maintenance, and extensibility advantages of high-level programming. These modules are not part of ELF, and they are not integrated into the operating system.

First, a pair of processes was created which send and receive packets to and from the PRU device previously integrated into ELF. The send process and receive process share a common address space which contains the connection table: this holds all status information for every connection being maintained by the station. These two processes are responsible for handling the PRN connection protocol described previously.

Within the same program module is a subroutine for creating a connection. This subroutine takes the address of the destination PRN device, consults the radio propagation connectivity matrix, and constructs the route which should prefix all packets sent to the specified destination. This route is stored as part of the connection status in the connection table when the connection is established.

The control process is another independent process for initializing the network and causing dynamic routing changes in response to changes in repeater-terminal propagation connectivity or repeater failures. It is responsible for keeping the propagation connectivity matrix up-to-date. This matrix is shared with the connection initialization routines which find a route to the requested device.

In the PRN initialization procedure, the control process calls on the connection module to establish a control connection to the station's PRU. It sends commands over this connection to trigger connectivity measurements (exploratory packets which request answerback from stations and repeaters within earshot). As measurement information comes back on this connection, the control process fills in entries in the connectivity matrix, and establishes control connections to the PRU's of the newly-discovered devices. This procedure is iterated until every station and repeater in the area has been configured into the network. At this point, the control process has an open control connection to every other station and repeater in the PRN.

When any repeater detects a significant routing event, e.g., failure of some previously established route or a request from a terminal to enter the network, the repeater forwards this information over its control connection to the nearest station. When a terminal comes on-line, the station establishes a data connection to it and provides an 'information service' to assist in completing the connection to the destination device, which may be either in the PRN or in the ARPANET. When repeater connectivity changes the station will update the connectivity matrix and reconfigure the network to bypass the failed link by modifying the routing control parameters in the effected repeaters.

The debug program is another independent process. On request from the maintenance terminal, it calls on the connection module to open a debugging connection to the PRU of interest. The debugger sends commands over this connection to examine or deposit words in the PRU's microprocessor memory, and the PRU responds with a positive acknowledgment for each command. There are also commands for setting traps on anomalous program conditions. When one of these conditions is encountered (assuming the PRU is still operational) it sends the appropriate trap code over the debugging connection to the debugger. This types out on the controlling terminal, which is presumably attended by a system programmer or operator.

The statistics collection module is another independent process which gathers data both by examining PRU memory, and by receiving statistics trap conditions spontaneously emitted by PRU's. This operation parallels the operation of the debugger described above.

Finally, the PRN TELNET process performs the second type of gateway function described above: conversion between the PRN connection protocol and the ARPANET host-host protocol. Terminals on the PRN appear identical to the terminals attached to the PDP-11, and are able to access remote ARPANET service hosts in the same way.

## CONCLUSION

Successful demonstration of the packet radio concepts will lead to new digital communications services for mobile terminals which are reliable, difficult to detect or jam, and which make efficient use of the electromagnetic spectrum by providing coexistence with current uses of the spectrum. Possible applications include both command and control communications and secure digital voice.

The system structure described above will provide reliable, efficient and maintainable support for packet radio network communications.

## REFERENCES

1. Kahn, R. E., *The Organization of Computer Resources into a Packet Radio Network,* these proceedings.
2. Abramson, N., R. Binder, F. Kuo and W. Okinaka, *ALOHA Packet* "Broadcasting, A Retrospect," these proceedings.
3. Garrett and S. Fralick, "A Technology for Packet Radio," these proceedings.
4. Kleinrock, L. and F. Tobagi, "Random Access Techniques for Packet Radio Networks," these proceedings.
5. Frank, H., R. Van Slyke and I. Gitman, "Packet Radio Network Design, System Considerations," these proceedings.
6. Fralick, S., D. Brandin, F. Kuo and Harrison, "Digital Portable Terminals," these proceedings.
7. Cerf, V. and R. Kahn, "A Protocol for Packet Network Intercommunication," *IEEE Trans. Comm., May 1974.*
8. Crocker, S., J. Heafner, R. Metcalfe and J. Postel, "Function Oriented Protocols for the ARPA Computer Network," *AFIPS Conference Proceedings,* Vol. 40.
9. Crocker, D. and J. Postel, *Remote Controlled Transmission and Echoing Telenet option.* RFC #581, Network Information Center.
10. Thomas, R. H., "A Resource Sharing Executive for the ARPANET," *AFIPS Conference Proceedings,* Vol. 42.