The DARPA Packet Radio Network Protocols

JOHN JUBIN AND JANET D. TORNOW, ASSOCIATE, IEEE

Invited Paper

In this paper we describe the current state of the DARPA packet radio network. Fully automated algorithms and protocols to organize, control, maintain, and move traffic through the packet radio network have been designed, implemented, and tested. By means of protocols, networks of about 50 packet radios with some degree of nodal mobility can be organized and maintained under a fully distributed mode of control. We have described the algorithms and illustrated how the PRNET provides highly reliable network transport and datagram service, by dynamically determining optimal routes, effectively controlling congestion, and fairly allocating the channel in the face of changing link conditions, mobility, and varying traffic loads.

I. Introduction

In 1973, the Defense Advanced Research Projects Agency (DARPA) initiated research on the feasibility of using packet-switched, store-and-forward radio communications to provide reliable computer communications [1]. This development was motivated by the need to provide computer network access to mobile hosts and terminals, and to provide computer communications in a mobile environment. Packet radio networking offers a highly efficient way of using a multiple-access channel, particularly with bursty traffic [2]. The DARPA Packet Radio Network (PRNET) has evolved through the years to be a robust, reliable, operational experimental network [3]. The development process has been of an incremental, evolutionary nature [4]; as algorithms were designed and implemented, new versions of the PRNET with increased capabilities were demonstrated. The PRNET has been in daily operation for experimental purposes for nearly ten years. In this paper we describe the current state of the DARPA PRNET.

We begin by providing a synopsis of the PRNET system concepts, attributes, and physical components in Section II. In Section III, we illustrate the mechanisms by which a packet radio automatically keeps track of a potentially continuously changing network topology. In Section IV, we describe the algorithms used to route a packet through the packet radio communications subnet. In Section V, we examine the protocols for transmitting packets. In Section VI, we describe some of the hardware capabilities of the packet radio that strongly influence the design and characteristics of the PRNET protocols. We conclude by looking briefly at some applications of packet radio networks and by summarizing the state of the current technology.

Manuscript received February 1, 1986; revised July 30, 1986. The work of J. Jubin was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract MDA903-85-C-0205. The work of J. D. Tornow was supported by the Defense Advanced Research Projects Agency of the Department of Defense under Contract MDA903-85-C-0254.

J. Jubin is with Collins Defense Communications, Rockwell International, Richardson, TX 75081, USA.

J. D. Tornow is with SRI International, Menlo Park, CA 94025, USA.

II. Description of the Packet Radio System

A. Broadcast Radio

The PRNET provides, via a common radio channel, the exchange of data between computers that are geographically separated. As a communications medium, broadcast radio (as opposed to wires and antenna-directed radio) provides important advantages to the user of the network. One of the benefits is mobility; a packet radio (PR) can operate while in motion. Second, the network can be installed or deployed quickly; there are no wires to set up. A third advantage is the ease of reconfiguration and redeployment. The PRNET protocols take advantage of broadcasting and common-channel properties to allow the PRNET to be expanded or contracted automatically and dynamically. A group of packet radios leaving the original area simply departs. Having done so, it can function as an autonomous group and may later rejoin the original network or join another group.

The broadcasting and common channel properties of radio have disadvantages too. These properties, for all practical purposes, prohibit the building of a radio that is able to transmit and receive at the same time. Therefore, the PRNET protocols must attempt to schedule each transmission when the intended PR is not itself transmitting. Also, transmissions often reach unintended PRs and interfere with intended receptions. Therefore, the protocols must attempt to schedule each transmission when the intended PR is not receiving another PR's transmission.

B. Automated Network Management

The PRNET features fully automated network management. It is self-configuring upon network initialization, reconfigures upon gain or loss of packet radios, and has dy-

0018-9219/87/0100-0021$01.00 © 1987 IEEE

namic routing. The network operator simply has to ensure that the packet radios are deployed so that every packet radio is situated so as to have at least one other packet radio within line-of-sight, and then turn the radios on; the packet radio is intended to operate unattended. Once installed, the system discovers the radio connectivity between packet radios and organizes routing strategies dynamically on the basis of this connectivity. After initialization, most communication networks maintain a static topology. A unique feature of the PRNET is the ease with which network topology can be altered without affecting the user's ability to communicate. Although RF connectivity is difficult to predict and may abruptly change in unexpected ways as mobile packet radios move about, the automated network management procedures used in the PRNET are capable of sensing the existing connectivity in real time and then exploiting this connectivity in order to continuously transport data and control packets, all in a way that is totally transparent to the users.

C. Network Components

The PRNET system comprises:

- The PRNET subnet, which consists of the packet radios. The PRNET subnet provides the means of interconnecting a community of users.
- The collection of devices (host computers and terminals), each attached to a packet radio via a wire high-level data link control (HDLC) interface, that wish to exchange data in real time.

The primary component of the packet radio communication network system is the packet radio. The generation of PR that is supporting the current research is designed to support the development and evaluation of advanced packet networking concepts and techniques. This packet radio equipment has been designated the Low-cost Packet Radio (LPR) [5], Fig. 1. The LPR consists of both digital and RF subsystems. It is capable of omnidirectional, spread-spectrum, half-duplex transmission/reception at 400- and 100-kbit/s data rates. It supports pseudo-noise code modulation of the information bits and forward error correction. The digital subsystem controls the routing and flow of packets between PRs while the RF subsystem transmits and receives packets over the radio channel. The PR receives packets of data either from its wire interface or from the radio channel. Each PR is responsible for receiving a packet and relaying it on to a PR that is one hop closer to the final destination. The packets can be routed either to another PR over the radio channel or to an attached device (i.e., host computer or terminal) via the wire interface. The protocols that run in the PRs encompass the Physical, Data Link, and Network layers, as defined in ISO's Open Systems Interconnection Reference Model [6].

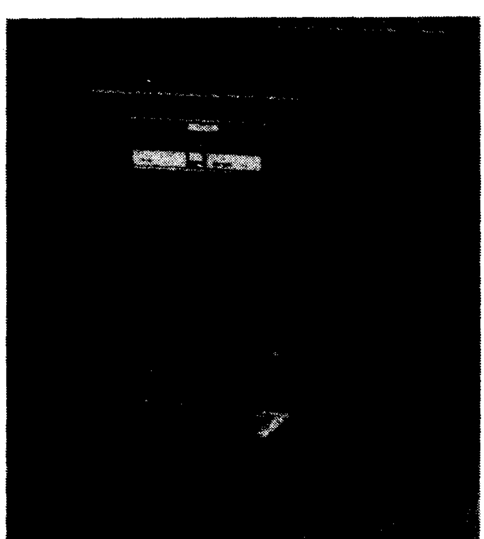


Fig. 1. Low-cost Packet Radio.

It is the local broadcast nature of the packet radio that gives the PRNET its unique networking characteristic: a PR's transmission is received by all PRs within line-of-sight; e.g., in Fig. 2, a connecting line indicates which PRs are within line-of-sight of each other. Since PRs *L*, *N*, and *Q* are all within line-of-sight of *M*, a transmission by *M* can be received by all of these PRs; they are said to be one hop away from *M*. Note, however, that a transmission by, say, PR *P* can be received only by PRs *L* and *N*. In general, a PRNET consists of many PRs that are not all within line-of-sight of each other, and packets must traverse multiple hops to reach their destination.

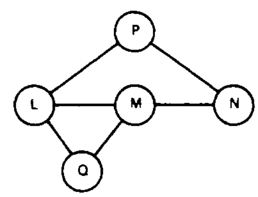


Fig. 2. Small packet radio network.

In order for a user to send data across a PRNET, a device (such as a small host computer) must be connected to a packet radio via an HDLC wire interface. Since the DARPA PRNET is a part of the DARPA Experimental Internet System [7], the devices are responsible for running the DoD-standard internetwork-, transport-, and application-level protocols (IP, TCP, and TELNET). These protocols ensure that the end-to-end communication between hosts is reliable and robust, and allow hosts on the PRNET to communicate with computers on various other packet-switched satellite, terrestrial, radio, and local area networks that also participate in the DARPA Internet. A host computer may be directly interfaced to a PR. If a user wishes to send data across the PRNET from a terminal or host that does not run the required protocols, a Network Interface Unit (NIU) [8], Fig. 3, may be used between the terminal or host and the PR.

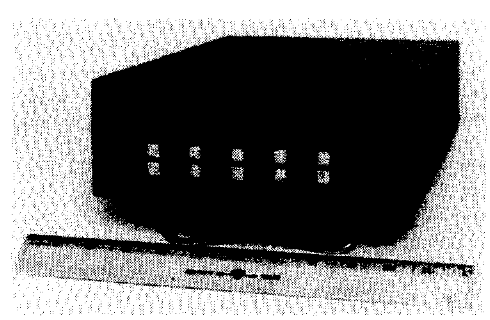


Fig. 3. Network Interface Unit.

The NIU performs the necessary host-to-host and terminal-specific protocols. The PRNET can also be accessed from other networks via an Internet gateway. In addition to supporting data, the PRNET protocols can also support a limited amount of digitized speech. Fig. 4 illustrates the PRNET with user devices as a part of the Internet. Note that, organizationally, the devices lie outside the PRNET subnet: the network appears as a black box providing packet communication service between pairs of user devices.

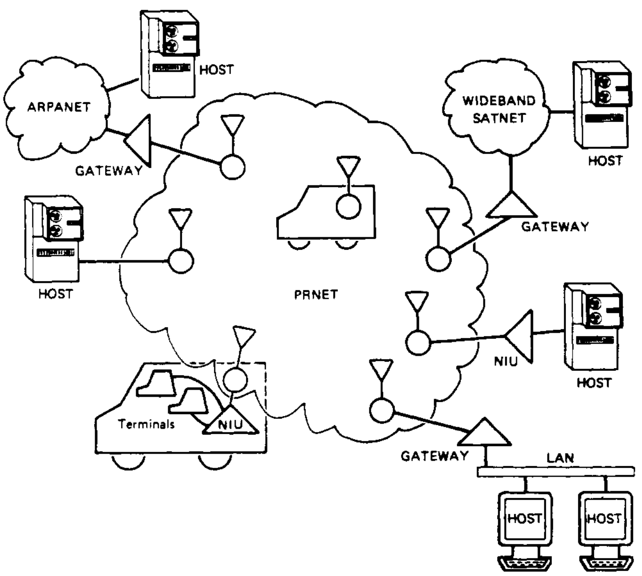


Fig. 4. Packet radio network in the Internet.

D. Network Size

The current network protocols are designed to support 138 entities, which can be any combination of PRs and attached devices, in a single network. For example, a typical network may be composed of 50 packet radios, many of which have attached devices and some of which serve only as repeaters. Multiple packet radio networks operating on different frequencies may be interconnected via Internet gateways to support applications which require larger networks. Research is currently being conducted to design algorithms which allow a single network to organize and manage thousands of packet radios without Internet gateways. An accompanying paper in this issue [9] describes this and other future research of the DARPA PRNET program more completely.

As the number of PRs increases, the number of neighbors (those packet radios in direct line-of-sight, described in Section III) per PR is likely to increase. The current protocols allow each packet radio to support up to sixteen neighbors. If however, packet radios are sited more densely, creating neighborhoods of more than sixteen radios, the congested PRs employ an algorithm to determine their "logical" neighborhood, the effect of which is to limit to sixteen the number of "actual" neighbors that congested PRs track. The primary decision criterion of this algorithm is to exclude the prospective neighbor with the most neighbors in common. The algorithm is careful to prevent isolated cliques, that is, radios which are not able to communicate with the rest of the network. Therefore, the network dynamically accommodates network topologies that result in congested neighborhoods. However, performance is likely to suffer when "actual" neighborhoods are large. Enhancements to the protocols which will prevent degradation of performance in large neighborhoods (such as adaptively setting dynamic link-level transmission parameters) is another subject of the DARPA PRNET program's future research.

E. Loading, Debugging, and Monitoring

The protocol software and replacements ("patches") to operating-system firmware in the PR can be loaded into the PR from a terminal via an RS-232 interface. A plethora of terminal commands—display memory, alter memory, select/display trace breakpoint, display state, execute jobs, execute procedure, etc.—are available for local debugging.

Protocol software and firmware patches can also be loaded from an already loaded and running neighbor PR via the radio channel. New versions of software will be automatically distributed thoughout the PRNET via neighbor loading.

PRs can also be loaded and debugged from a remote host computer. All the commands available to a local operator are available to a remote operator. The remote operator can load/debug not only a PR that is executing the regular protocol software, but also a "frozen" PR, that is, a PR that has stopped executing the regular program because of a self-detected fault.

Another host computer function called the Network Monitor is used to aid in observing and analyzing the PRNET. Each PR continuously gathers measurements on bidirectional link quality, nodal capacity, and route characteristics. The Network Monitor will collect data from each of the packet radios and display them graphically to aid the network designers in characterizing and understanding the network behavior. The Network Monitor is currently defined as an optional component of the network because it does not contribute in any way to the management of the network.

III. Link Connectivity and Route Calculation

The PRNET features fully distributed network management [10]. Each packet radio gathers and maintains enough information about network topology so that it can make independent decisions about how to route data through the network to any destination, even before it is given a packet to deliver or forward. The network information is stored in three tables:

- neighbor table
- tier table
- device table.

These tables are established by the packet radio upon initialization, and are updated automatically by the PRs as the topology changes. In the following paragraphs, we describe how the tables are established, maintained, and used.

A. Neighbor Table

After a PR (say PR L in Fig. 2) has been powered on, and has loaded its protocol software into RAM, it begins the process of establishing and maintaining local connectivity. As long as the PR is powered up, it will broadcast a Packet Radio Organization Packet (PROP) every 7.5 s, announcing its existence and information about the network topology from its perspective. However, when PR L is first powered up, it does not know anything about the network so its PROP simply announces its existence. The PROP is received by all neighboring radios (those in direct connectivity with L). The set of radios that hear the PROP note the event in their neighbor tables. Similarly, PR L will hear the periodic PROPs of its neighboring PRs and will start to build its own neighbor table. Once PR L begins participating in the network by passing data, the data packets are used in addition to the PROPs to maintain the neighbor table.

In addition to maintaining a list of the PRs that are one hop away, the neighbor table tracks the bidirectional quality of the link to/from those neighbors. Each PR keeps a running count of the number of its transmissions and includes this count in its PROP. This value is used by the receiving PRs to evaluate the quality of the link. The link quality is measured in terms of the ratio of number of packets correctly received from the transmitting PR during a PROP interval to the number of packets that the transmitting PR actually transmitted during that same interval. For example, if PR L transmits 80 packets in a 7.5-s period and PR M receives 50 of them correctly, the L-to-M link quality is $\frac{5}{8}$ (just high enough for a "good" rating). Once the calculation has been completed, the receiving PR zeros out the count of the number of packets received from the transmitting PR and stores the running count of transmissions by the transmitting PR, so that if a PROP is missed the calculation will still be valid. Continuing the example, once PR M has determined that the link quality is $\frac{5}{8}$, it will set the number of packets received from L to zero, and store '80' as the last count of packets transmitted by L. Then, say PR M receives another PROP from PR L which indicates that PR L's running transmission count is now 180. PR M subtracts 80 from 180 to get the transmission count since the last computation. If PR M has received 90 packets since the last computation, the new L-to-M link quality is $\frac{9}{10}$.

Quality values are smoothed, and hysteresis is applied when switching between good and bad ratings. Separate values are maintained for the 400- and 100-kbit/s data rates. This neighbor table, then, represents a PR's accumulated knowledge about all the PRs that are one hop away. If a PR moves, its new connectivity is dynamically determined by the same mechanism.

B. Tier Table

Routing in the PRNET is done by having each PR maintain knowledge of the best PR to forward packets to every prospective destination. This is similar to the early ARPANET routing algorithm [11]. Each PR that receives PR L's PROP notes that it is only one radio hop away, so it is at "tier 1" with respect to PR L, and includes this information about the network topology in its next PROP. Still other PRs hear the PROPs broadcast by the PRs which are at tier 1 with respect to L, note that they are one hop away from the PRs from which the PROP was received, and therefore at tier 2 with respect to PR L (which emitted the original PROP). These second-tier PRs, in turn, relay this information out another tier in the PROPs, and so on. For example, in Fig. 2, PRs P, M, and Q learn that they are at tier 1 with respect to PR L after receiving PR L's PROP. The next time that PRs P, M, and Q broadcast their PROPs, they include the information that they are at tier 1 with respect to L. When PR N receives PR P's PROP, it knows that it is at tier 1 with respect to PR P, and that it is at tier 2 with respect to PR L via PR P. Thus tier information ripples outward from each PR at an average rate of 3.75 s per hop, eventually reaching all the PRs in a tiered ring fashion. In this way, ultimately every radio knows its distance in tiers from any prospective destination PR and which PR is the next PR enroute to that destination PR. This information is maintained continuously by each PR in its tier table, as shown in Fig. 5.

TIER TABLE		
Destination PR	Next-PR In Route	Tier
N	N	0
M	M	1
P	P	1
L	M	2
Q	M	2

Fig. 5. Typical tier table for PR N.

The goal of the tier table is always to maintain the "best" information about how to get to a destination packet radio. The "best" route is currently defined as the shortest route with good connectivity on each hop. Therefore, the tier table is updated only when certain conditions are met. Let us consider that N has received a PROP from M, and examine the conditions for which N updates its tier table, where the update for any given destination PR consists of storing:

- as the next PR enroute: M
- as the tier: M's reported tier, with respect to the destination PR, incremented by one (referred to as 'T' below).

A necessary condition is that M be a bidirectionally good neighbor of N. In addition to the good neighborliness, any one of the following conditions must be met for N to update its tier table:

- There is no stored tier data for that destination PR.
- The prospective new tier ('T') is strictly less than the stored one.
- M is the same PR that is already stored as the next PR.

When the link quality to a neighboring PR (say the link from PR N to PR M) becomes "bad," all routes in N's tier table for which the neighbor PR M is the next PR in the route are also set bad. This means that M can no longer provide a reliable way for N to send packets to a destination PR, and a new next PR should be chosen, and thus a new, good route can be formed even if it is longer than the old, bad one. Just as good tier data are spread via PROPs, the news of bad tier data is also spread via PROPs. PR N will include the news in its PROP that it can no longer provide a route to the destination PRs that it reached via M. It is important to spread the bad news in an orderly fashion. For example, suppose PR M is the next PR with respect to destination PR L in PR N's table. Assume that M no longer has connectivity with L, so it marks its tier data bad and broadcasts this information in its next PROP. If N does not receive the PROP for some reason, then N will continue to announce in its PROP that it has tier data to L, and M will assume that it can get to L via N. Mechanisms exist in the protocols to break these route loops and to ensure that obsolete data do not override the up-to-date data.

Information gathered from the headers of data packets is also used to update the tier tables. Under heavy traffic loads, this greatly improves the network's responsiveness to a rapidly changing topology, and contributes significantly to the PRNET's automatic reconfiguration capability.

C. *Device Table*

The PRNET provides a dynamic addressing capability. The DARPA packet radio network's device-to-PR mapping is known as logical addressing. With logical addressing, the relationship of a device to a PR may be altered as necessary. So that all packet radios can become aware of all the attached devices connected to the subnet, each such device periodically sends a control packet across the wire interface to its attached PR. The PR notes the existence of the attached device and includes it in its PROP to be propagated to the other radios in the network at an average rate of 3.75 s per hop. Device-to-PR correspondence information is also learned from headers of data packets. Through these mechanisms, ultimately all packet radios know of all devices and to which packet radios the devices are attached. If a device is moved from one PR to another, or if it goes down, its new status is made known to the network in the same manner. This device-to-PR correspondence information is stored by each packet radio in a device table.

In Fig. 6, Device 1 launches a packet destined for distant Device 2. The packet is sent across the wire interface to PR

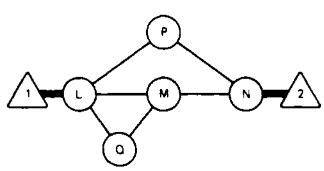


Fig. 6. Packet radio network with attached devices.

L, which uses its device table to map Device-id 2 to its attached PR (*N*). The device-to-PR mapping is totally transparent to the user and to the device. PR *L* then uses its tier table to initiate the routing of the packet through the packet radio communications subnet towards PR *N*. *N*'s device table has Device 2 stored as an attached device, so once the packet has arrived at *N*, it is passed over the wire interface to Device 2. If the Device 2 is no longer attached to *N* when the packet arrives, the packet will be discarded by *N*. It would be a violation of the layering architecture to recompute the address of the destination end device while routing at the subnet layer. However, the PRNET's logical addressing facilitates the rapid dissemination of Device 2's new address. Consequently, when the end-to-end reliable protocol, TCP, initiates a retransmission of the packet (having failed to receive an acknowledgment), the network is likely to be able to route the packet to the new address of Device 2.

In addition, the PRNET offers generic logical addressing. Generic addressing allows a device to request service from any other device in a general category, e.g., any name server or any gateway. The packet is addressed to an ID that may belong to more than one device. The network will deliver the packet to the closest device with the requested generic ID.

IV. Forwarding Protocols

Despite the local broadcast nature of the packet radio, a user packet is not flooded throughout the PRNET to get to its destination because multiple-hop flooding would use too much of the finite capacity of the network's common channel. Instead, generally speaking, a packet traverses a single path through the network, and is acknowledged at every packet radio along the path (Section IV-B). Multiple-path forwarding may occur only if alternate routing has to be employed (Section IV-D). Forwarding is accomplished via information read from the device and tier tables (Sections III-B and III-C) and from the packet headers (Section IV-A). Forward error correction increases the probability of error-free reception, and cyclic redundancy checksums prevent, with high probability, the forwarding of packets that are not error-free (Section IV-C).

A. *Packet Headers*

Every packet transmitted by every PR contains several headers, which add about 10 percent to the packet length. Each header corresponds to a protocol layer. Strict layering produces a clean, structured design, but also causes the headers to be longer than they could be because of duplication of some fields and fragmentation of others among the headers. The packet headers that are of concern to this paper are the end-to-end (ETE) header and the routing header.

ETE Header: The ETE header is created by the source device. It contains the source device ID, which is used to update the PRs' stored device–PR correspondence data (Section III-C), and the destination device ID, which is used in forwarding (Section IV-B). Also in the ETE header is a type-of-service flag that the source PR transfers to the routing header to customize forwarding for low delay/reliability applications such as speech (Section V-D). The ETE header stays on the packet from its creation by the source device throughout its forwarding through the PRNET including its delivery to the destination device.

Routing Header: The routing header is created by the source PR, encapsulating the ETE header. Its fields that are of concern to this paper are:

Routing Header Field	Protocol Function Used In	Section Described In
Source PR ID	acknowledgment	IV-B
	alternate routing	IV-D
Sequence number	acknowledgment	IV-B
	alternate routing	IV-D
Speech type-of-service flag	transmission	V-D
Previous PR ID	acknowledgment	IV-B
Previous PR's transmit count	pacing	V-A
Transmitting PR ID	acknowledgment	IV-B
Transmitting PR's transmit count	pacing	V-A
Next PR ID	forwarding	IV-B
	acknowledgment	IV-B
	pacing	V-A
Lateral alternate routing flag	alternate routing	IV-D
Alternate routing request flag	alternate routing	IV-D
Tier	alternate routing	IV-D
Destination PR ID	forwarding	IV-B
	alternate routing	IV-D

The routing header stays on the packet throughout its forwarding through the PRNET subnet. The source PR ID, sequence number, and destination PR ID created by the source PR stay fixed throughout the packet's journey to the destination PR. The rest of the fields are updated by every intermediate packet radio. The routing header is stripped off the packet when it is delivered by the destination PR to the destination device.

B. Forwarding-Acknowledgment-Retransmission Protocol

Forwarding: Packets are forwarded over a single path through the PRNET by each packet radio using the information in the packet's headers and in its own device and tier tables. Each PR uses this information, first, to decide whether it should be the one to transmit the packet on, second, to update the routing header before transmitting the packet on, and third, to update its own tables.

By example, let us enumerate the steps in fowarding a packet through the network of Fig. 6, from Device 1, through PRs *L*, *M*, and *N*, to Device 2:

```
Device 1
        Sets, in ETE header, destination device ID = 2.
        Wire-transmits the packet.
PR L
        Receives packet.
        Sets, in routing header:
          destination PR ID  = N (from device table)
          previous PR ID     = φ (signifies an attached
                                  device)
          transmitting PR ID = L
          next PR ID         = M (from tier table)
          tier               = 2 (from tier table)
        Radio-transmits the packet.
PR M
        Receives packet.
        Determines, from next PR ID = M in routing
          header, that it should process the packet.
        Sets, in routing header:
          previous PR ID     = L
          transmitting PR ID = M
          next PR ID         = N (from tier table)
          tier               = 1 (from tier table)
        Radio-transmits the packet.
PR N
        Receives packet.
        Determines, from next PR ID = N in routing
          header, that it should process the packet.
        Determines, from destination PR ID = N in rout-
          ing header, that packet is at the destination PR.
        Determines, from destination device ID = 2 in ETE
          header and from device table, that packet
          should be wire-transmitted.
        Wire-transmits the packet.
        Sets, in routing header:
          previous PR ID     = M
          transmitting PR ID = N
          next PR ID         = φ
          tier               = φ
          Radio-transmits the routing header (done for
          acknowledgment purposes, explained in the
          next section).
Device 2
        Receives packet.
```

Other PRs within range also receive each transmitted packet. For example, *L*'s transmission is received not only by *M* but also by *P* and *Q*. Since neither *P* nor *Q* is the next PR in the routing header, they both discard the packet. The packet transmitted by *M* is also received by *L* and *Q*; it is discarded by *Q* and processed by *L* as a passive acknowledgment as explained in the following section.

Acknowledgment: Let's look at the same sequence of events from the perspective of just the packet transmissions:

```
Device 1     PR L      PR M      PR N    Device 2
        wire
     +------>
                  radio
                +------>
                        radio
                  <------+------>
                                       wire
                                     +------>
                               radio
                             <------+
```

Notice that PR *M*'s single radio transmission is received not only at the next PR, PR *N*, but also at the previous PR, PR *L*. Thus the single transmission not only forwards the packet on to the next PR, but also acknowledges to the previous PR reception of the packet by the transmitting PR. This kind of acknowledgment, which takes advantage of the broadcast nature of packet radio, is known as a passive acknowledgment. Since the destination PR *N* wire-transmits the packet to forward it to the destination Device 2, PR *N* must also radio-transmit a separate acknowledgment to acknowledge reception of the packet to PR *M*. This kind of acknowledgment, which consists of only the routing header of the packet, is known as an active acknowledgment.

A packet that is received at a packet radio qualifies as an acknowledgment if it meets two criteria:

- It is a duplicate of the original, indicated by its source PR ID and sequence number matching those of the original packet.
- It has made it farther downstream toward the destination, indicated by one of several criteria, including:
  - It, or perhaps an alternate-routed (Section IV-D) copy from another PR, got to the intended next PR—indicated by its transmitting PR ID being the same as the original packet's next PR ID.
  - It came from this PR—indicated by its previous PR ID being the same as this PR's ID.
  - It or a copy got to a PR that at least thinks it is closer to the destination PR—indicated by its tier being smaller than the original packet's.

Retransmission: If a PR, say *L*, that has forwarded a packet does not receive an acknowledgment within a certain interval, it retransmits the packet. *L* transmits the packet as many as six times waiting for an acknowledgment and then finally discards the packet (depending on the transport-level protocol to end-to-end retransmit as is required to provide the desired end-to-end reliability). The interval between retransmissions is determined by the pacing protocol (Section V-A) and grows with each packet that is not acknowledged by a given next PR, say *M*. When the interval becomes too large, *L* sets its connectivity rating with *M* as nonexistent, without waiting for a much longer time to time out on the absence of any packets from *M*.

C. Error Control

Forward Error Correction: The LPR uses long convolutional codes and sequential decoding to perform forward error correction (FEC). The current software employs a coding rate of one-half (with provision for future enhancement to variable rate coding). Half-rate coding doubles the transmission time and sequential decoding increases the processing time significantly, but FEC often prevents retransmitting (described in the previous section), thus often preventing even more delay and processing in environments with high bit error rates.

Cyclic Redundancy Checksum: The LPR also has hardware to generate a 32-bit cyclic redundancy checksum (CRC). This provides error detection with a very low undetected error rate. Received packets that fail the CRC test, even if they have been sequentially decoded, are simply discarded and therefore treated as if they had not been received at all. The packet can be checksummed by sections with individual CRCs. The current software checksums a forwarded packet's header and text separately. This allows a PR that receives the header correctly (after FEC) to use it as a passive acknowledgment and to update its neighbor table and tier table, even if the text has uncorrectable errors.

D. Alternate Routing

If a PR has not received an acknowledgment after three transmissions of a packet, it sets the alternate routing request flag in the packet header to request forwarding help on the last three retransmissions. In some respects, the last three retransmissions are treated the same as the first three. For example, they are all separated by the same retransmission pacing interval. Also, if the desired next PR receives one of the last three transmissions, it forwards the packet on as usual, even though the alternate routing request flag was set. However, a PR whose ID is not the next PR ID in the header also accepts such a packet for forwarding if its tier table shows the tier to the distination PR to be less than, or equal to, the tier in the header.

Returning to the example from Fig. 6 and Section IV-B, let us say that PR *P* receives a packet transmitted by PR *L* whose destination PR ID is *N*, whose tier is 2, and whose next PR ID is *M*. Let us also say that the packet's alternate routing request flag is set. In this case, *P* will accept the packet for forwarding, even though it is not *M*, because its stored tier to *N* is 1, which is less than the tier of 2 in the packet header. PR Q will also accept the same packet for forwarding because its stored tier to *N*, 2, is the same as the packet header's. To prevent a packet from being alternate routed at the same tier forever in a circle around the destination PR, the lateral alternate routing flag in the header is set when this "lateral" alternate routing is in progress, prescribing that the next helper (assuming alternate routing help is still needed) must be strictly closer to the destination.

E. Duplicate Filtering

An unnecessary retransmission occurs when the next PR is slow in forwarding or when the transmitting PR does not successfully receive the next PR's acknowledgment. Whenever an unnecessary retransmission occurs, the next PR may receive a duplicate of a packet it has already queued for forwarding or has already forwarded. Alternate routing may also create duplicate packets in the network, as shown in the example in the preceding section, namely at PRs *P* and *Q*. Such duplicate packets are not desirable because their forwarding wastes some of the finite capacity of the network's common channel. Therefore, received packets that otherwise would be forwarded but that are recognized as duplicates are filtered out instead of being forwarded.

Packets are uniquely identified for the purpose of duplicate filtering in the same way that they are identified to qualify as acknowledgments (Section IV-B), namely, via the packet header's source PR ID and sequence number. These two entities are together known as the unique packet identifier (UPI). The UPI of a newly received packet about to be queued for forwarding is compared to the UPIs of 1) packets that are currently queued for forwarding and 2) the last few packets that have been forwarded and acknowledged; any match causes the new, duplicate packet to be discarded. An active acknowledgment is transmitted to prevent the transmitter of the duplicate packet, that is the previous PR, from retransmitting unnecessarily any more, in the following circumstances. An active acknowledgment is always transmitted in the second case above, since the original packet has already been acknowledged and therefore will no longer be (re)transmitted and thereby provide acknowledgment to the previous PR. For the same reason, if an acknowledgment to the queued packet is received before the packet has had a chance to be (re)transmitted in the first case above, an active acknowledgment is also transmitted.

V. Transmission Protocols

Section IV described how a PR decides that a received packet should be forwarded and how it selects the next PR to which the packet should be forwarded. After those decisions are made, the packet is then put in the radio transmit queue according to a protocol which promotes "fairness" in the use of the PRNET resources (Section V-B). The time at which it is selected to be transmitted is determined by a three-component pacing protocol (Section V-A). The transmission time may be further delayed slightly by the carrier sense multiple access protocol (Section V-C). Transmission parameters are chosen according to the measured link quality and to the type-of-service desired by the user (Section V-D).

A. Pacing

The pacing protocol [12] provides flow and congestion control while contributing to the efficient but fair use of the radio channel. There are three interrelated components to pacing: single-threading, measuring the forwarding delay through each neighboring PR, and applying a function of the measured delay to separate (re)transmissions to each neighbor PR.

Single-Threading: Single-threading requires that a packet transmitted to a certain next PR be acknowledged (or discarded after six transmissions) before another packet is transmitted to the same PR. Recall from Section IV-B that the acknowledgment is generally provided passively by the same transmission that forwards the packet, the effect being that the acknowledgment is not transmitted until the PR is ready to accept another packet from the same previous PR. (The special kind of acknowledgment, the active acknowledgment, is not transmitted until after the packet has been forwarded in all cases.) The net result is that congestion is deflected as much as possible away from a network bottleneck back to the source PR. In addition, there is a limit of two buffers allotted per PR for packets received over the wire interface for entry into the PRNET. This reflects the congestion completely out of the subnet. Thus single-threading provides a form of flow control as well as congestion control.

Forwarding Delay: To get an indication of how much delay should be applied between packet (re)transmissions, each PR measures the forwarding delay of each packet that it forwards through each neighbor PR. A PR, say L, measures forwarding delay by recording the time at which its transmission completes and then subtracting that recorded time from the time at which the reception of the acknowledgment from the next PR, say M, of the original packet completes. (The propagation time is negligible compared to the transmission time.) If L had to retransmit the packet, it knows from which of its retransmissions to measure (actually, estimate) the time of transmission, because the acknowledgment's header contains the previous PR's (L's) transmit count, which is a record of the transmitting PR's (L's) transmit count when the packet was actually received by M. If no acknowledgment is received, the entire time that a PR waited for the acknowledgment since the first transmission of the packet is used as the forwarding delay measurement.

The forwarding delay encompasses all the processing, queuing, carrier-sensing/randomization (described in Section V-C), and transmitting delays which a neighbor PR experiences from all its own activity and from the activity going on in its neighborhood before it completes the transmission of the acknowledgment. Forwarding delay measurements are exponentially smoothed and stored on a per-neighbor PR basis. Smoothing the measurements maintains a short-term history of the delay through each neighbor PR, which is presumed to be valid for the next transmission/acknowledgment cycle. Furthermore, when coupled with single-threading, the forwarding delay of the highest delay PR along a forwarding path is reflected back to the source PR.

Pacing Function: Consider the worst case scenario for radio channel use: a sequence of maximum-length packets transmitted at the slower 100-kbit/s data rate with half-rate FEC. For this scenario, the processing delay is small compared to the transmission/reception delay. Furthermore (returning to the example from Section IV-B), the transmitting PR, L, must allow time for the next PR, M, not only to receive L's transmission and to forward it on but also to receive the acknowledgment from its (M's) next PR, N. In other words, the packet radio's forwarding cycle has three frames; no PR, generally speaking, can transmit more than one-third of the time. The three-frame nature of the forwarding cycle is illustrated in Fig. 7.

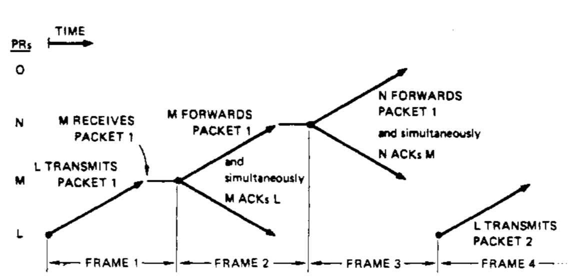


Fig. 7. Three-frame packet forwarding.

In recognition of this free-frame nature, the pacing protocol applies a multiplier of three to the measured/smoothed forwarding delay, "D," to separate transmissions of different packets and retransmissions of the same packet to that neighbor PR. Note in Fig. 7 that PR L waits to transmit packet 2 until three times the D measured through PR M has elapsed since it (L) transmitted packet 1. This same conservative algorithm is applied in the case of 400-kbit/s transmissions, even though the actual transmission time is a less significant part of the total D in the case of 400-kbit/s transmissions. For the case of packets just entering the PRNET from an attached device, the separation between transmissions is a more complicated function of D: it is slightly greater than the three times D (the formula for radio-received packets) for small values of D, and then increases proportional to the square of D. The purpose is to increase the choking effect rapidly as the perceived network congestion increases.

In summary, pacing adapts PRs' times of transmission to provide congestion control in the face of a bottleneck PR, but on the other hand, uses the channel efficiently when there is no bottleneck.

B. "Fairness" Queuing

As a general rule, packets are handled first-in first-out (FIFO). However, there are two exceptions to the FIFO rule. The first exception is a result of the pacing protocol described in the preceding section. Pacing can slow the rate of forwarding to one neighbor PR compared to the rate to another neighbor PR. Therefore, a packet j headed in a "fast" direction and therefore ready to go, in the pacing sense, is transmitted before one i headed in a "slow" direction and therefore not ready to go, even if i arrived before j. There is, however, a single radio transmit queue, so that, when several packets are ready to go at the same time, they will be transmitted FIFO, that is, i before j.

The second exception to the FIFO rule is that received packets are not always entered at the tail of the radio transmit queue. The purpose of this exception is to overcome the inherent unfairness that may exist among various receive links. Consider in Fig. 8 three streams of packets merging at one PR (X), all going to the same neighbor PR (Y): one stream is coming over a weak radio link from PR A, another is coming over a strong radio link PR B, and the last is coming over the practically error-free wire interface from device C. With straight FIFO queuing, the wire inter-

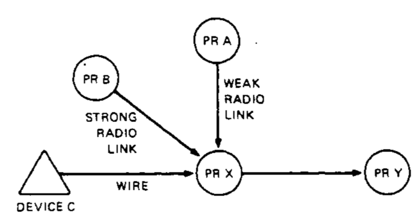


Fig. 8. LPR with variable-strength receive links.

face stream would monopolize PR X's buffer and transmission resources, and the weak radio link would not get its share. Furthermore, the pacing protocol would provide positive feedback to further exacerbate the situation.

When faced with a monopolizing receive link that has more than one of its packets destined for the same neighbor PR already in the radio transmit queue, the "fairness" algorithm takes a packet that arrives over a different receive link, destined for the same neighbor PR, and places it in the queue ahead of the second packet from the monopolizing link. Two examples of this "butting in line" in the interest of fairness are shown in the following scenario based on Fig. 8. (The "butting" packets are packets 5 and 6.)

Packet arrival event at PR *X* (all packets destined for PR *Y*)	PR *X*'s transmit queue <–head tail–>
Packet 1 arrives from PR *A*	1*A*
Packet 2 arrives from PR *B*	1*A*, 2*B*
Packet 3 arrives from device *C*	1*A*, 2*B*, 3*C*
Packet 4 arrives from device *C*	1*A*, 2*B*, 3*C*, 4*C*
Packet 5 arrives from PR *B*	1*A*, 2*B*, 3*C*, 5*B*, 4*C*
Packet 6 arrives from PR *A*	1*A*, 2*B*, 3*C*, 6*A*, 5*B*, 4*C*

C. Carrier Sense Multiple Access

The pacing protocol described in Section V-A prescribes intertransmission delays that foster noninterfering transmission patterns. However, irregularly spaced user traffic is bound to cause more than one PR to schedule transmissions which overlap in time. One problem with two PRs transmitting at the same time is that, if their transmissions are intended for each other, neither will be received, since the packet radio is half-duplex. Another problem with overlapping transmissions is that they could interfere, or collide, with each other at the PR(s) that can receive both transmissions.

The carrier sense multiple access (CSMA) protocol attempts to prevent a PR from transmitting at the same time a neighbor PR is transmitting [13]. A PR can tell if a neighbor PR is transmitting by reading its hardware indication "bit-synchronization-in-lock." This condition is also described by the phrases "the channel is busy" and "a carrier is being sensed." Whenever a carrier is being sensed, a PR will refrain from transmitting.

Even though PRs can carrier-sense, there is still a latent problem if more than one PR, say PR *L* and PR *M*, have transmissions being held off by carrier-sensing the transmission of another PR, say PR Q. Let us say that *L* decides to transmit as soon as it senses the reception from Q complete. The problem is that there is a finite delay from the time when *M* senses the channel idle after its reception from Q completes until the time when it first senses the channel busy from the transmission from *L*; if *M* believed the bit-synchronization-in-lock indication during this interval, it would end up overlapping transmissions with *L*. The components of this delay interval are listed below and their relationship is shown in Fig. 9:

- the time from when the reception from Q completes at *M* until the same reception completes at *L*, due to the difference in propagation time from Q to *M* compared to that from Q to *L*;
- *L*'s receive-to-transmit switchover time;
- the propagation time from when *L* starts the actual transmitting until *M* starts receiving;
- *M*'s bit-synchronization-in-lock acquisition time, also known as bit-synchronization detection time.

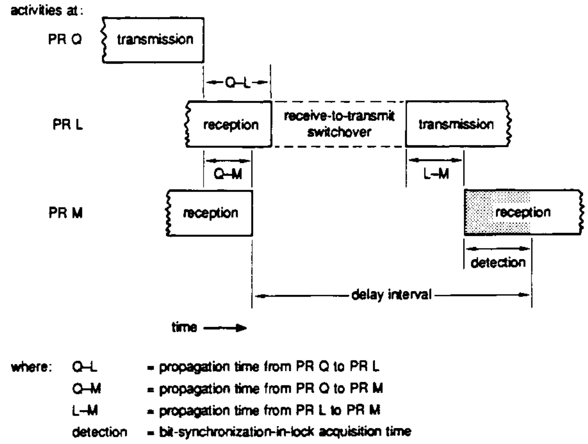


Fig. 9. CSMA delay interval.

The CSMA randomization delay protocol to decrease the probability of overlapping transmissions, then, is as follows:

1) If the channel is busy:
   a) monitor the channel until it goes idle;
   b) select a random number and delay for that number of the delay intervals defined above;
   c) if the channel is (still or again) busy, go back to step a).
   Otherwise...
2) Transmit the packet.

If two or more PRs are unfortunate enough to select the same smallest number in step 1b), there will be simultaneous transmissions. The more likely occurrence is that only one PR selects the smallest number. That PR, say *L*, will transmit, and the rest of the PRs will wait long enough to be able to sense that the channel is busy again. Those PRs that are in connectivity with *L* will be able to carrier-sense and refrain from transmitting. Those PRs that are not in connectivity with *L*, also known as "hidden PRs," will not be able to sense *L*'s transmission and may themselves also transmit; however, the randomization delaying will decrease the degree of overlap of the transmissions, thus helping the "capture" effect (Section VI-B) to allow the first transmission to succeed. The range from which the random number is selected increases with each retransmission that fails to elicit an acknowledgment, to further decrease the probability of collision and to further increase the probability of capture.

Notice that no CSMA randomization delay at all is applied if the channel is initially sensed idle. This is exactly what is desired for the most frequently encountered case of the first transmission of a packet by a given PR. If a CSMA delay were applied in this case, it might work against the pacing protocol attempting to determine the appropriate time to transmit and actually cause PRs' transmissions to unnecessarily overlap. However, there are two cases when an initial CSMA randomization delay is applied no matter what the initial state of the channel is:

- when a PR accepts for forwarding a packet that was requesting alternate routing—to avoid several PRs responding simultaneously to the request;
- on retransmissions, that is on transmissions of a packet

whose first transmission has not been acknowledged—to get PRs whose previous transmissions of a packet have collided out of sync with each other, so their retransmissions do not also collide.

For these two cases, the protocol defined above is still applied, but starts with step 1b) instead of 1a).

D. Transmission Parameters

For most users' packets, standard default values are used for most of the parameters affecting their transmission: total number of retransmissions, number of retransmissions before alternate routing is requested, the maximum value of forwarding delay. Data rate is the only parameter that is selected dynamically for most users' packets. The 400-kbit/s data rate is favored to minimize transmission time; the 100-kbit/s rate is selected only when the link quality to the next PR is bad at the 400-kbit/s rate but good at the 100-kbit/s rate.

One special type-of-service is offered to users. It provides lower delay, and thus higher throughput, but less reliability and was designed primarily for packetized speech. A user can invoke speech type-of-service on a packet-by-packet basis via the speech type-of-service flag in the ETE header. A comparison between the values of transmission parameters for speech type-of-service versus regular service is shown below:

	Type of Service	
	Regular	Speech
Data rate (kbits/s)	400, unless 100 is the only good link quality	400 always
Total number of transmissions	6	3
Number of transmissions before alternate routing is requested	3	1
Maximum forwarding delay ("D")	858 ms	30 ms

VI. Hardware Capabilities

This section discusses in more detail the LPR's hardware capabilites: the Data Link Layer functions introduced in Section IV and V, and the Physical Layer functions not yet introduced. This section also presents the parameters and measurements available to the software, including some that are not currently being used, and discusses potential ways in which they can be used. A more detailed discussion of the LPR hardware is in an accompanying paper in this issue [14].

An overview of the LPR's hardware functions is given in Fig. 10. The figure traces a packet through the stages leading up to radio transmission and following reception. It also shows the hardware parameters available for the software to set at each transmit and receive stage and the measurements provided to the software at each receive stage.

A. Error Control

Forward Error Correction: As discussed briefly in Section IV-C, the LPR uses long convolutional codes generated on the transmit side and sequential decoding on the receiving side to perform forward error correction (FEC). There are three coding rates—7/8, 3/4, and 1/2—with constraint lengths 91, 63, and 36, respectively. They provide increasing degrees of gain—up to 9 dB at 1/2 rate—at the expense of increased channel and decoding overhead. Applying an error correcting code will increase the probability of packet's successful reception and the link throughput. On the other hand, when the noise and interference levels are low, the high overhead involved in a low-rate code will tend to decrease the link throughput since most of the packets would be received correctly without FEC. Thus highly dynamic algorithms must be used to estimate the channel's current condition and to determine the corresponding best selection of FEC.

The LPR's FEC hardware also provides to the software the option to select 2-bit soft decision decoding (the bits actually being generated during demodulation and then passed on to the sequential decoder). Soft decision increases error correcting power in a Gaussian noise environment at the expense of greater decoding overhead. The current software employs 1/2-rate coding with hard decision. The effect of burst noise is mitigated by interleaving (that is, shuffling the bit order) on the transmit side and then de-interleaving on the receive side; the net result is that burst noise gets spread across 128 bits.

Error Detection: After error correction on the receive side, the LPR provides error detection via a 32-bit cyclic redundancy checksum (CRC). On the transmit side the CRC is generated before FEC encoding.

B. Modulation

Symbol Modulation: As discussed briefly in Section V-D, there are two selectable data rates, 400 and 400 kbits/s for symbol modulation by the body of the packet. The 400-kbit/s rate is used whenever possible to minimize channel usage.

TRANSMISSION		RECEPTION		
Function	Parameters	Function	Parameters	Measurements
CRC generation	packet sectioning	CRC checking (error detection)		
FEC (convolutional encoding)	encoding rate	FEC (sequential) decoding (error correction)	hard/soft decision	FEC error count
interleaving		deinterleaving		
preamble generation	preamble contents, whether to time-tag	bit sync detection, frame sync detection and		time of arrival
symbol modulation	data rate	symbol demodulation	hard/soft decision	signal + noise, noise, multipath
chip modulation (PN encoding)	key, time slot duration or time slot itself, qualifier	chip demodulation (PN encoding)	key, time slot duration, table of qualifiers	
up-conversion	frequency	down-conversion	frequency	
adaptive power control	attenuation	automatic gain control		receive power (AGC) level
transmission		reception		

Fig. 10. Stages of packet transmission and reception.

However, the 100-kbit/s rate provides an additional 6-dB receive processing gain over the 400-kbit/s rate. Furthermore, the 100-kbit/s rate provides more resistance to multipath due to longer symbol time.

Chip Modulation: One of the LPR's most important features is the direct-sequence spread-spectrum waveform produced by chip modulation at a rate of 12.8 megachips per second using a pseudo-noise (PN) sequence. Spreading provides a processing gain over noise and interference. The gain is approximately 13 dB at the 400-kbit/s data rate and 19 dB at the 100-kbit/s data rate. The particular PN code sequence generated is a function of the 56-bit key, the 25-bit slot time, and a 32-bit entity that we call the "qualifier." For a packet to be detected at all, the code sequence on the receive side must match that on the transmit side, and thus all three parameters of which the code sequence is a function must also match. The PN-code-generation algorithm is the National Bureau of Standard's Data Encryption Standard (DES) [15].

Even when the slot time and the qualifier are fixed, the generated PN code changes with each bit of the packet body. This bit-by-bit code changing can provide "capture," the successful reception of a packet transmitted from one PR even when overlapped by the transmission of a packet from another PR at least one preamble time later [16]. Letting the slot clock run makes the code generated a function of the slot time at the beginning of the transmission producing "code slotting" and a time-varying waveform. The time slot duration can be as small as 5.12 ms. For a code-slotted packet to be detected at all, the receive side must employ the same slot time as the transmit side did when it transmitted the packet. Code slotting requires the network to be synchronized so that transmitting and receiving PRs' slots coincide to a large degree. Synchronization accuracy is provided by the hardware capability to "time-tag," that is, to read the time to a resolution of 5 μs at a particular point, called "frame synch," in the transmission/reception of the preamble. On the transmit side, the time is put in the packet being transmitted. On the receive side, the time is provided along with the packet itself (containing the transmit time-tag) to the software.

PN qualifiers work as follows: The transmit side puts the index to a commonly known table of qualifier values into the preamble of the packet and then PN-encodes the body of the packet using the qualifier value itself; the receive side then fetches the index from the preamble and then the value from its own table and uses it to PN-decode the body. Using various PN qualifiers would provide code division multiple access (CDMA). The qualifier could be the receiving PR's ID, in which case receiver-directed CDMA would be provided. Or the qualifier could be a "group" ID; a network could be divided into groups to mitigate the adverse effects of overdenseness, or swaths of PRs could be grouped into "superhighways."

C. *RF Stages*

RF Conversion: There are 20 RF frequencies between 1718.4 and 1840.0 MHz selectable in the LPR. This capability can be used to set up colocated but noninterfering PRNETs. The LPR is frequency-agile to the extent that the hardware provides to the software the capability to change the frequency on a packet-by-packet basis with only a 3-ms settling time after a change. This feature can be used to provide frequency division multiple access (FDMA).

Adaptive Power Control: The hardware provides the software the capability to select, on a packet-by-packet basis, attenuation to the nominal 5-W transmitted signal of 0 to 24 dB in 8-dB steps. This feature can be used to decrease or increase network connectivity or to match the signal power of transmissions from various-distanced PRs at each receiver, to mitigate the "near–far" problem.

D. *Receive Measurements*

Several measurements on the receive side are provided to the software: receive power (AGC), signal + noise, noise, multipath, and FEC error count. These can be used to better quantify the link qualities between neighboring PRs (Section III-A) that provide the basis for the PRNET routing algorithms (Section III-B).

These receive measurements can also be used to better decide which parameters/values to employ on the transmit side of the link. For example, when appreciable multipath is present, the 100-kbit/s data rate should be used. Also, a low signal-to-noise ratio, as calculated from signal + noise and noise measurements, can be improved by a combination of higher power (that is lower attenuation), lower FEC coding rate, and lower data rate.

The problem is that changes in parameters/values to improve the performance of one link tend to degrade the performance of other links: Higher transmit power increases the range and the contribution of interference; lower FEC coding rate and lower data rate each increases the length of time of interference. Such network self-interference can be mitigated by employing CDMA and/or FDMA. Here the main issue to be explored in future work is dynamic subnetwork organization in the face of changing connectivity.

VII. Conclusion

The success of the DARPA packet radio program has led to the investigation of packet radio applications in both the military and commercial arenas. Experimental packet radio networks have been established by both the Army and the Strategic Air Command for purposes of determining the applicability of PR technology to support distributed, survivable command, control, and communications [17]. The military has also initiated an investigation of the potential for extending the capabilities of existing military point-to-point radios with packet appliques. A packet applique is a set of processors (such as a personal computer) which is interfaced to existing radio equipment. The applique constitutes the digital portion of the packet radio node; thus the protocols that control the routing and flow of packets among nodes are implemented in it. The enhancement of the radio equipment by the addition of the packet applique greatly increases the network's flexibility by supporting automatic communication among nodes that are not in direct connectivity with one another.

Another application of packet radio networks currently being explored is the requirement for providing distributed access to either distributed or centralized information. Library automation is a good example of this need. As technology advances, more and more interlibrary information is being exchanged electronically. Until now, very little progress has been made in obtaining fixed-cost telecommunication services for libraries. Except for expensive and specialized facilities (such as private microwave) that are beyond the budgetary reach of virtually all libraries, one can expect to pay common carriers every month for services used, and common-carrier rates have been steadily increasing. Packet radio, however, offers libraries the potential for purchasing data communication capabilities at a reasonable fixed cost.

Single-hop packet radio networks are also being used to improve the efficiency of commercial operations. For example, the use of a small hand-held radio with a limited keyboard is being experimented with by restaurants. The waiters, bartenders, and cooks are all equipped with a packet radio. The waiter enters an order and its destination (either the bartender or the cook), then waits to receive a packet indicating that the order has been completed.

In summary, packet radio is an exciting technology that is beginning to play an important role in the local distribution of information. In this paper we have presented the current state of the DARPA packet radio network. The primary component of the PRNET is the LPR. The LPR has many sophisticated features that can provide enhanced flexibility in designing a robust and reliable packet-switched communications network. Fully automated algorithms and protocols to organize, control, maintain, and move traffic through the PRNET have been designed, implemented, and tested. By means of these protocols, networks of about 50 packet radios with some degree of nodal mobility can be organized and maintained under a fully distributed mode of control. We have described the algorithms and illustrated how the PRNET system (i.e., the LPRs along with their attached devices) provides highly reliable network transport and datagram service, by dynamically determining optimal routes, effectively controlling congestion, and fairly allocating the channel in the face of changing link conditions, mobility, and varying traffic loads.

Acknowledgment

Many of our colleagues working on the packet radio program have contributed to the ideas presented in this paper. In particular, we would like to acknowledge contributions from DARPA, CECOM, BBN Communications, Hazeltine Corporation, Rockwell International, and SRI International.

References

[1] R. E. Kahn, "The organization of computer resources into a packet radio network," *IEEE Trans. Commun.*, vol. COM-25, no. 1, pp. 169–178, Jan. 1977.

[2] R. E. Kahn, S. Gronemeyer, J. Burchfiel, and R. C. Kunzelman, "Advances in packet radio technology," *Proc. IEEE*, vol. 66, no. 11, pp. 1468–1496, Nov. 1978.

[3] J. Jubin, "Current packet radio network protocols," in *INFOCOM'85 Proc.*, Mar. 1985.

[4] K. Klemba *et al.*, "Packet radio executive summary," Tech. Rep. prepared for Defense Advanced Research Projects Agency by SRI International, July 1983.

[5] D. Behrman and W. C. Fifer, "A low-cost spread-spectrum packet radio," in *MILCOM'82 Proc.*, 1982.

[6] H. Zimmermann, "OSI reference model—The ISO model of architecture for Open Systems Interconnection," *IEEE Trans. Commun.*, vol. COM-28, no. 4, pp. 425–432, Apr. 1980.

[7] B. Leiner, R. Cole, J. Postel, and D. Mills, "The DARPA Internet Protocol Suite," *IEEE Trans. Commun.*, vol. COM-23, no. 3, pp. 29–34, Mar. 1985.

[8] D. Beyer, M. Lewis, and J. Tornow, "Network interface unit," Tech. Rep. SRNTN 35, SRI International, Sept. 1985.

[9] N. Shacham and J. Westcott, "Future directions in packet radio architectures and protocols," this issue, pp. 83–99.

[10] J. Westcott and J. Jubin, "A distributed routing design for a broadcast environment," in *MILCOM'82 Proc.*, 1982.

[11] J. M. McQuillan and D. C. Walden, "The ARPA Network design decisions," *Comput. Networks*, vol. 1, pp. 243–289, Aug. 1977.

[12] N. Gower and J. Jubin, "Congestion control using pacing in a packet radio network," in *Proc. IEEE Millitary communication Conf. (MILCOM'82)*, vol. 1, pp. 23.1-1–23.1-6, Oct. 1982.

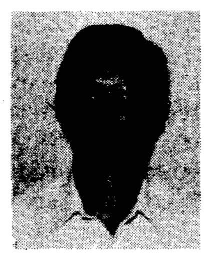
[13] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, Dec. 1975.

[14] W. C. Fifer and F. J. Bruno, "The low-cost packet radio," this issue, pp. 33–42.

[15] Fed. Inf. Process. Stand., *Data Encryption Standard*, publ. 46 ed., Nat. Bur. Stand., Washington, DC, 1977.

[16] D. H. Davis and S. A. Gronemeyer, "Performance of slotted ALOHA random access with delay capture and randomized time of arrival," *IEEE Trans. Commun.*, vol. COM-28, no. 5, pp. 703–710, May 1980.

[17] M. S. Frankel, "Advanced technology testbeds for distributed, survivable command, control and communications," in *Proc. IEEE MILCOM Conf.*, vol. 3, Oct. 1982.



John Jubin was born in Philadelphia, PA, on March 25, 1947. He received the B.S.E. degree in electrical engineering from Princeton University, Princeton, NJ, in 1968, and the M.S. degree in operations research from Southern Methodist University, Dallas, TX, in 1977.

From 1969 to 1973 he was a communications-electronics engineer in the U.S. Air Force Communications Service. In 1973 he joined Texas Instruments, Dallas, TX, where he developed software for real-time seismic data processing systems and for Global Positioning System user sets. In 1977 he joined Rockwell International's Collins Defense Communications organization, where he has worked primarily on DARPA's Packet Radio and Survivable Radio Networks (SURAN) programs. He is currently principal investigator for Rockwell on SURAN. He has (co-)authored four papers on packet radio network protocols for IEEE conferences. He has designed packet-switched communications protocols also for other systems, including the Ground Wave Emergency Network.

Janet D. Tornow (Associate, IEEE) received the B.S. degree in mathematics from Douglass College, Rutgers University, New Brunswick, NJ, in 1977, and the M.S. degree in operations research from Stanford University, Stanford, CA, in 1979.

In 1978 she worked at Bell Laboratories. She joined SRI International in 1979, where she is a Research Engineer. Her primary responsibility is principal investigator for SRI on the DARPA Survivable Radio Networks program. Her research interests include the specification and development of packet radio protocols, the integration of packet radio network systems, and network characterization and performance evaluation.

Fig. 1. Low-cost Packet Radio.

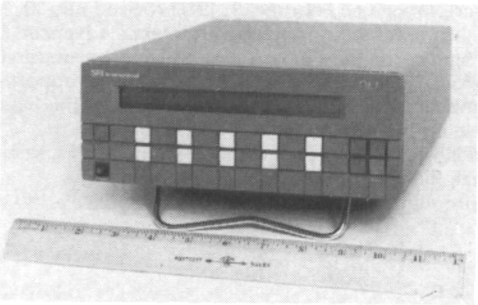


Fig. 3. Network Interface Unit.