2

**Bolt Beranek and Newman Inc.**

LEVEL III A088 039 (2)

**Report No. 4340**

# Command and Control Related Computer Technology: Packet Radio

## Final Report

Including Quarterly Progress Report No. 20
1 September to 30 November 1979

December 1980

Prepared for:
Defense Advanced Research Projects Agency

(9) Quarterly progress rept. no. 20 (Final) 1 Sep-30 Nov 79

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER<br>BBN Report No. 4340 | 2. GOVT ACCESSION NO.<br>AD-A093 135 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>(6) COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY: PACKET RADIO. | | 5. TYPE OF REPORT & PERIOD COVERED<br>Final/Quarterly Progress Report 1Sept79-30Nov79 |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>(10) M. Beeler, J. Westcott | (15) | 8. CONTRACT OR GRANT NUMBER(s)<br>MDA903-75-C-0180,<br>ARPA Order-2935 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Bolt Beranek and Newman Inc.<br>50 Moulton Street<br>Cambridge, Massachusetts 02138 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>(12) 38 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Advanced Research Projects Agency<br>Information Processing Techniques Office<br>1400 Wilson Blvd. Arlington, VA 22209 | (11) | 12. REPORT DATE<br>Dec~~ember 1~~980 |
| | | 13. NUMBER OF PAGES<br>ii + 35 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)<br>(14) BBN-4340 | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED. IT MAY BE RELEASED TO THE CLEARINGHOUSE DEPARTMENT OF COMMERCE FOR SALE TO THE GENERAL PUBLIC.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Packet Radio, Computer Communications, PDP-11, TCP, Gateway

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document describes progress on development of a packet radio network. Activities reported include work on Station Software and Internetworking Research and Development.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

JOB

BBN Report No. 4340

December 1980

# COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY:

## Packet Radio

Final Report and Quarterly Progress Report No. 20

1 September 1979 to 30 November 1979

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935 Contract No. MDA903-75-C-0180

## TABLE OF CONTENTS

# 1. INTRODUCTION

An important component of the Packet Radio project is the station software, providing a variety of control, coordination and monitoring functions. BBN's role in developing this software is to specify, design, implement and deliver programs which perform these functions.

Progress during this quarter centered on Labeler process enhancements, gateway efforts, and extensive support activities. Each of these is covered in a section devoted to those subjects. Additional sections cover meetings, publications, negotiations, other station work, and hardware. A special section deals with network testing, which would ordinarily appear under "Labeler" or "Support", but is treated separately this time due to an extensive series of mobile run experiments it covers.

Besides being the twentieth Quarterly Progress Report, this is the final report on contract MDA 903-75-C-0180, covering a period of five years of design, development, implementation, refinement and delivery of operating Packet Radio station software and related Internetwork gateways and Transmission Control Program. (Other efforts, on Speech Compression and Vocoded-Speech Evaluation, originally a part of this contract, were split off into separate contracts in 1977.)

Over this period we have published a total of 59 technical notes (Packet Radio Temporary Notes), as listed in the appendix. While Quarterly Progress Reports have briefly described the content of the notes for the quarter covered, it is beyond the scope of this report to discuss all these technical notes in depth; the reader is referred to the Quarterly Progress Reports and to the PRTNs themselves. In many cases PRTNs have anticipated network design needs and guided development toward a more reliable, efficient network with expanded capabilities. Other PRTNs have provided documentation of software designs and operator interfaces. Some of these, notably 212 (on measurement file entries) and 174 (on the Labeler process) have undergone numerous revisions and redistributions.

The effort covered by this contract has resulted in design, implementation and delivery of Packet Radio station software capable of controlling and coordinating a network of Packet Radios, now in use at Fort Bragg, SRI, and the Collins Radio and BBN testbeds. Another result is the design, implementation and delivery of Transmission Control Protocol and the Internet Protocol on which TCP depends, now in use at BBN, SRI, and ISI, with parallel implementations elsewhere (UCL, MIT), which permit reliable data transmission across multiple, interconnected networks. And third, this effort has resulted in design, implementation and delivery of gateway software by which networks are interconnected. These three components act in concert to provide access from radio-linked terminals at Fort Bragg, through the Fort Bragg Packet Radio network to the Fort Bragg gateway, and from there through the ARPANET to host services at ISI. The internet (TCP and gateway) aspects of this development are already covered in the Quarterly Reports and in separate Internet group reports (Internet Working Group reports and Internet Experimenters' Notes), but the development of the Packet Radio station and the protocol it uses are covered in a special retrospective chapter of this report.

## 2. QPR 20: MEETINGS, TRIPS, PUBLICATIONS

### 2.1 Meetings and Trips

During this quarter BBN representatives attended the Internet meeting in London the first week of September, and the CAP 5 Implementers' Meeting at SRI September 25-28. We also met with another BBN group, headed by Dr. John McQuillan, to discuss control of routing and congestion in the PR net. His group advised that no conclusions could be reached on so complex a topic without an investigation and analysis in depth. Tentative advice, however, was as follows.

1. The PR net is over constrained, and the most promising constraint to work for relaxation of is the small amount of memory in PR's.
2. Alternate routing as now implemented sounds questionable, if not definitely a poor reaction to hop transport trouble.
3. Delay is one reasonably promising measure to indicate congestion, but what control measures to take once congestion is detected is a complex and difficult problem.
4. Estimating the overall capacity of a packet switching node is a significant, multidimensional task.

### 2.2 Publications

During this quarter we distributed several technical papers on both internet and Packet Radio net-specific subjects, as described in the paragraphs below.

Internet Experimenters' Note (IEN) 109, "How to Build a Gateway"

This paper explains the detailed technical concepts behind implementation of a gateway, an internet packet switch which connects two or more networks using Internet Protocol.

Packet Radio Temporary Note (PRTN) 278, "New PRN Device ID Policy"

This paper specifies a new interpretation of PR network 16-bit device ID numbers, slightly modified from the old policy, to permit the proper recognition of gateways operating outside the station.

IEN 120 and PRTN 279, "Internet Routing and the Network Partition Problem"

A difficult problem arises when a network separates into two or more partitions, each fully functional within itself, but unable to reach any nodes in the other partition(s) except indirectly through the internetwork (catenet). This includes subproblems of addressing, packet switch and host status detection, and suppression of routing loops. This paper presents a routing design which specifically addresses network partitioning.

PRTN 280, "Transfer Points"

This PRTN specifies the detailed design of transfer points, which allow PR net routes with greatly expanded length. Present routing allows only seven "hops" from PR to PR. The transfer point routing will also support multiple stations, which are planned for implementation in the coming months.

PRTN 281, "Multistation Design Specification"

After distribution of PRTN 280 (see above), we received no objections from the PR Working Group community. Consequently we proceeded with this note, a final description of multiple station operation.

## 2.3 Negotiations and Informal Documents

### 2.3.1 Checksumming assessment

During this quarter we also participated in a discussion by network mail on possible "checksumming" algorithms for future use in the PRNET. The two main candidates involved were right rotate or left rotate by one bit, then one's complement add. We timed optimally-coded loops of each of these, executing in a PDP-11/40 station. The right rotate was:

```
         ;<R0> = 0
         ;<R1> = Address of first word to include checksum
         ;<R2> = Number of words to include
CKSM:    CLC
         ROR R0
         BCC CKSM1
         BIS #100000, R0
CKSM1:   ADD (R1)+,R0
         ADC R0
         SOB R2,CKSM
         ;<R0> = Resultant checksum
```

The right rotate executes in 15 microseconds per pass. The left rotate was:

```
CKSM:    ADD (R1)+,R0
         ADC R0
         ASL R0
         ADC R0
         SOB R2, CKSM
```

This uses the same argument and result registers, and executes in 10 microseconds per pass. Either algorithm can be modified to perform the add and the rotate in the reverse order, at no cost in execution speed. On the IPR's TI9900 processor, it appears that a left rotate, which would probably be called a "shift left circular" if the TI9900 instruction set had one, can be constructed from the sequence:

| | |
|---|---|
| SLA | (shift left arithmetic) |
| JNC | (jump if no carry) |
| INC | (increment) |

### 2.3.2 Repair of broken routes

During this quarter SRI conducted and reported some measurements of IPR performance. These results, and discussion they prompted, made it clear there was uncertainty in the PR community about the mechanism of achieving a new point-to-point route assignment after a link breaks. We described this process to clarify the understanding of the net's performance among the Packet Radio Working Group members. A slightly improved version of that communication is reproduced here.

1. The link breaks. Start timing from this point. Assume that the link changes from good quality to solidly broken, so that suddenly no more packets will get through. (If link quality was previously of poor quality, the PR would probably have previously attempted to send a link failure PDP to the station. If link quality was previously of good quality but degraded only to poor quality, not a complete break, then some packets would get through but not all. If three consecutive LROPs were lost on the poor link, the scenario would be the same as described below. If at least one of every three consecutive LROPs got through, the link quality as measured by the PR would decline until it passed the threshold for being a good neighbor. This would then be reported to the station in a PDP. See Collins' CAP5.1 documentation for details of this and related PR functioning. We assume a sudden, complete break of the link because it is simpler to discuss and is what was done in the lab tests.)

2. 50 seconds must elapse since an LROP was heard over the link. 50 seconds is chosen as being slightly greater than three LROP intervals, which are 15 seconds each. An LROP may have been received anywhere from 15 seconds before the link broke to just before it broke. Thus a delay of from 35 to 50 seconds is incurred for the PR to decide the link has vanished. [Incremental time, 35 to 50 seconds; total elapsed time, 35 to 50 seconds.] (There is a chance that the PR might be so busy that it does not get around to noticing that no LROP has arrived over the broken link for 50 seconds, until somewhat more than 50 seconds has elapsed. We shall neglect this possibility.)

3. The PR decides the link has vanished, that is, the neighbor is no longer a neighbor. The PR now tries to send a PDP to the station announcing a change in goodness of a neighbor. Reasons the PR might not be able to send such a PDP essentially immediately, and their effects, are as follows.

   a. Another PDP has been sent within 30 seconds. The PR remembers the neighbor change and sends a PDP when the 30 second interval has elapsed. We assume the network was stable before the link broke, so no PDP had been sent, so no delay is incurred.

   b. No buffer is available in which to compose the PDP, even using the "scrounge" routine to steal buffers off the radio transmit queue. The PR will send the PDP when a buffer becomes available. Delay is indeterminate, but we assume small and we assume that this can only happen with extreme congestion, so we ignore this effect.

   c. Congestion may be so severe that each time the PDP is queued for radio transmission, it is deleted by the "scrounge" routine before actually appearing on the radio channel. Again, although this could result in indeterminate delay, we assume it will not occur at the low to moderate traffic level we are discussing, so we ignore this effect.

   Thus the PDP reporting the neighbor goodness change should be transmitted essentially immediately.

[Incremental time, 0 seconds; total elapsed time, 35 to 50 seconds.]

4. The PDP travels through the PR net and is delivered to the station. Congestion can slow this delivery, or even stop it altogether; but we assume that the level of congestion, if any, is not that severe. [Incremental time, under 1 second; total elapsed time, 35 to 50 seconds.]

5. The PDP is delivered to the label process in the station, and processed by it. This can be affected by the following conditions.

   a. The connection process may have recently accepted a PDP on its sole listening connection, thus binding that connection to the PR which sent the PDP. When the PDP of concern arrives, no connection would be available, and the connection process would drop the PDP. The connection process would print, "dropped packet -- no connection". The labeler would soon reopen another listening connection, however, and end-to-end retransmissions of the PDP (or network-generated duplicates which the PR net didn't filter out) which arrived after this would match this new connection and be delivered. Thus we assume only occasional delay, and rather slight delay even then, due to this mechanism. In the near future, the labeler will have multiple listening connections to reduce this problem even further.

   b. The labeler may be busy, such as recomputing the routing matrix. Until it is free, it will not service the newly arrived PDP. The interval during which the labeler is occupied is assumed small, however; and because we assume network stability prior to the link break, the labeler would be at most processing a PDP with no matrix recomputation required.

   c. The labeler may have had a connection to this PR open, which was closed within the last 20 seconds. The current implementation of the connection process, to fix the "oscillation" bug in SPP design, will ignore packets from that PR to the labeler until the 20 seconds has elapsed.

> If a substantial part of the 20 seconds remains, the PR will get no acknowledgement, discard its PDP, and transmit a distress LROP. This will cause neighbors of the PR to each send a PDP to the station with "neighbor in distress" as a reason. This would cause the labeler to mark the link as bad and the PR as needing relabeling. The labeler would then relabel the PR, through a different link if the broken link remains broken, in the near future. Meanwhile, the PR with the broken link will be sending link failure PDPs to the station, as in several of the alternative cases above. We assume that this 20-second timeout is not likely to influence the case at hand, partly because of the assumption of a stable net and partly because the only such contending connection in a stable net would be the servicing of a maximum interval PDP, generated because no PDP had been generated by the PR in question for 30 minutes. The expected delay from this cause is 1/90 second on the average. In the future, one of two proposed protocol modifications (simplex SPP2, or flagging obligatory open/close packets) is likely to be adopted and implemented, eliminating this 20-second deaf period.

Thus we assume that negligible delay occurs in the labeler reacting to the neighbor goodness change PDP. [Incremental time, under 1 second; total elapsed time, 35-50 seconds.]

6. Further traffic routed over the broken link is not hop acknowledged. If the traffic were to coincidentally stop just when the link were broken, no new route would be assigned for further traffic on the failed route. In our case, however, we assume the traffic is rather constant, so the link failure (failure to obtain a hop acknowledgement from a bad neighbor on a packet's specified route) is noticed with very little delay after step (3), in which the PR decided the neighbor was bad. Note that steps (4) and (5) are going on in parallel with the current step (6). Step (6) also incurs a delay while the retransmissions are performed to the PR specified in the packet's route, but this delay is small also. [Incremental time, under 1 second; total elapsed time, 35-50 seconds.]

7. The PR tries to send a PDP to the station reporting the link failure. This is subject to all the same possible delays cited in step (3), but this time, delay (a) cannot be ignored. Instead, we know the PR has sent a PDP very recently. Therefore, almost all of the 30 second minimum interval between PDPs still remains. [Incremental time, 30 seconds; total elapsed time, 65-80 seconds.]

8. The link failure PDP traverses the net, is delivered to the labeler, and is processed by the labeler. These steps are similar to steps (4) and (5), and for the same reasons we again assume there is little delay incurred. [Incremental time, under 1 second; total elapsed time, 65-80 seconds.]

9. The labeler assigns a new point-to-point route for this traffic. The route assignment packet traverses the net from the station to the source PR. The source PR stores the new route in its route table. New packets entering the net are now given the new route. For many of the same reasons cited above, particularly those of a net without severe congestion, we assume the delay in these actions is short. A recomputation of the routing matrix in the labeler will be required. Combined with the transport of the PDP and the route assignment packet, we estimate the delay to be around a second. [Incremental time, about 1 second; total elapsed time, 66-81 seconds.]

The distribution of total elapsed time should be essentially flat over the 66-81 second range, leading to an expected value of 73.5 seconds. This compares well with the observed value of 72 seconds. During the interval, traffic will follow the less efficient alternate routing protocol.

In following up this discussion, we have identified another issue which seems to be the crux of the problem. This is the unnecessarily close coupling between the link fail PDP and the good neighbor bit. As we have suggested in the past, the entry for each neighbor in each PR's neighbor table should have an associated counter. This counter keeps track of successive transmit failures. After some number (probably in the range 5 to 15) of successive failures occur, the link is declared bad and reported to the station. Any successful transmission resets the counter associated with that neighbor. This scheme significantly increases the responsiveness to solid link failures. We believe there are a number of issues along these lines to be considered.

### 2.3.3 IPR Down-line Loading

Collins personnel expressed interest in eventually running IPR diagnostics remotely, and concern that the IPR Operating System will never all fit into PROM. These motivated them to propose a new design for down-line loading of IPRs, one which in particular supports the selection of the file to load from a set of several possible files. Their proposal contains three relatively minor changes to the protocol agreed upon at the September 25-28 meeting:

1. The station may fragment IPR code anywhere, not just at line boundaries.
2. Dollar sign delimits an IPR load, instead of a double asterisk.
3. Any load packets with an odd number of bytes are filled with null to complete a word.

The Collins proposal also contains five major changes to the negotiated design:

1. The IPR specifies a file name of up to 8 characters, including an ETX delimiter, in load request LROPs and PDPs.
2. No version number in the file name indicates the latest (highest numbered) version known to the station.
3. Null file name (ETX only) indicates the current default CAP protocol.

4. The first load packet contains the file name of the file to which the PR's request was matched.

5. Load request LROPs and PDPs also contain diagnostic error status for futures use in remote running of diagnostics, but which the station can ignore of now.

We reviewed and accepted the Collins proposal. Due to these late-proposed changes to the design tentatively finalized in the September meeting, and to the more complex service requirements, delivery of station support of IPR down-line loading has been rescheduled and cannot be completed before the termination of the contract.

### 2.3.4 Miscellaneous

In response to an SRI request, we assessed the possibility of again having the old measurement process back in the station, particularly for ease of remotely controlling traffic streams. It appears this will probably work, if there is room for the process in station memory.

We responded to a long-standing need for documentation of the format and use of Terminal-On-Packets (TOPs), written for users of the PRNET, by drafting a brief document. Comments from SRI and Collins, however, suggest that the format should be simplified in the near future. Therefore we have temporarily postponed issuing the document, pending resolution of the format question.

# 3. QPR 20: THE PACKET RADIO NETWORK

## 3.1 Labeler

During this quarter we installed several improvements in the Labeler:

- Printout of link quality command was improved to clarify packet flow direction.
- Commenting feature added so typescripts can be annotated; also provides communication between a remote XNET user and the station operator during debug and test sessions.
- Printout of new PRs, and the PR forwarding them, added; helps with hunt for bad IDs, which show up as new PRs.
- Operator control of point-to-point routing added by allowing operator to prohibit Labeler fixing of bad routes; this also helps in hunt for bad IDs.
- Installed timeout on device/PR correspondence; allows reconfiguration of device attachment.
- Routes are now stored internally as IDs, rather than indices; this makes route refreshing more efficient and solves a problem of duplicate IDs appearing in routes.
- Initial implementation of CAP 5.2 Labeler was completed; in particular, this version periodically requests source/ destination pairs from PRs in groups of 3 and, if routes are present, transmits the new best route.

## 3.2 Support

Four areas saw significant support efforts this quarter:

- Various versions of PR CAP software (5.1.2, 5.1.3, 5.2.1, 5.2.2 and 5.2.3) were placed on assorted disks at SRI, Collins and BBN.

- Several days were spent in support of the NTC demo; besides consultation, debugging and checkout, this required modifying the XNET debugger to load the COMSAT gateway, and modifying the gateways in general to support changes to the catenet configuration.
- The Collins station was successfully brought up on the ARPANET; diagnostics loaded from a disk specially prepared at BBN were run, and a cable connector wiring error was found.
- The BBN IMP10 interface driver module was updated, so BBNF can now be brought up on ARPANET host port 0, IMP 71. Some user programs, not working due to the "high" host number, were modified.

## 3.3 Transmission Control Protocol/Program

During this quarter considerable effort was expended in testing TCP version 4. Several problems were found and fixed, resulting in a TCP4 which is now supporting user programs routinely at BBN and ISI. The Internet User Queue facility was also extensively tested and is now supporting user applications.

After cooperating with SRI to identify several bugs in early versions of LSI-11 TCP4 from SRI, a working version emerged. This was converted to run in the station under ELF. Also, the version 4 TIU code was imported from SRI and used to build a program for the TCP test TIU here, "Altacoma".

## 3.4 Gateways

We released new versions of the gateways at SRI, UCL, NDRE and BBN. The gateway at UCL was formerly a gateway between the ARPANET and the SATNET; it is now a gateway between the UCLnet and the SATNET. The gateways at SRI, NDRE, and BBN were modified to eliminate the UCL gateway on the ARPANET from their tables of neighbor gateways.

The mini-gateways on the SATNET include code which implements the new gateway monitoring protocol. A program has been implemented on the BBNA TOPS-20 system which polls the gateways as specified in the new monitoring protocol to obtain status reports. This program periodically obtains counts of packets received and sent on the gateway's network interfaces and the distance and route from each gateway to each network. It also obtains a report from each gateway whenever a neighbor gateway or network interface is declared up or down. These reports are written onto a TOPS-20 file.

We now spend a very significant part of our time in maintaining and delivering gateway systems. We currently support 4 ELF based gateways on Packet Radio nets, 3 ELF based gateways on the SATNET, 2 mini-gateways on Packet Radio nets, and 4 SATNET mini-gateways. We are responsible for delivering gateway software to 1 disk at UCL, 2 disks at Fort Bragg and 5 disks at SRI.

## 3.5 Hardware

A failure of BBN PRs was first thought to be a software problem in CAP5.2, but was traced to very poor radio reception, in one direction only, over the coax link between the two EPRs. Diagnostic results were sent to Collins. Collins personnel came to BBN and made various repairs and alignment adjustments. The PRs are now operating correctly.

The PTIP host port to which PR station number 2 is connected failed this quarter, and has been repaired.

## 4. QPR 20: NETWORK TESTING

During this quarter we participated in some further network performance testing, taking a leading role in designing, advising during execution, and analyzing the results. The tests themselves, consisting of "mobile runs" because they included the mobile PR and terminal in the van at SRI, were carried out by SRI staff.

In previous runs, the experimental testbed had produced so much congestion that no meaningful conclusions could be drawn. The factors contributing to congestion which were eliminated from the runs this quarter are:

1. The gateway used to be resident in the station, but is now exported to a separate gateway machine.
2. Excessive and relatively uncontrolled traffic, such as user traffic from Xerox, printout of a large text file ("superprogrammer") from an ARPANET host via TCP, and several traffic streams, was eliminated. Only known, controlled test traffic was employed in the new runs.
3. Traffic often traversed the station PR in the past, but now user traffic travels direct point-to-point routes between terminal PR and gateway PR. This avoids congesting the station PR to a large extent.
4. SRI enjoys the continuous monitoring of network "health" provided by having the station connection process packet printer turned on to print a summary of every packet processed. The casual use of this debugging and monitoring aid, especially when used in a busy network, has caused delay and backing up in the station. During these new runs, the packet printer was enabled only for printing error or problem situations, not for all packets.

In addition, many of the runs were performed on the Packet Radio net alone, eliminating the effects of Internet TCP connections. These effects confuse results because they are not controllable or easily measurable by the PR net experimenter. They include:

1. Serious, varying, unpredictable delays due to load average fluctuations on the ARPANET host machine (SRI-KL or an ISI host).
2. TCP retransmissions which are often invoked by delays due to congestion, and which aggravate that congestion.

The mobile runs in which we participated this quarter were held on September 4, 6, 18 and 28. Typically, at least two different runs were performed on each day. Some conclusions of long range import from the September 4 run are repeated here:

> We believe that disabling packet printing, except for the abnormal messages, solves the problem of the station connection process frequently dropping packets because of having no buffers. This is evidenced by the complete absence of "dropped packet - no buffers" printout during these runs.
>
> The recovery from a congested situation (which is what we suppose caused the problem in run 2 of intermittent outages and zero link qualities to stationary, good repeaters) seems linked to alternate routing in a negative way. It appears that alternate routing continues to get some portion of the offered traffic through while congestion occurs, until finally, measured link qualities drop to zero (or to some low value). Then the PRs will no longer attempt to alternate route packets over these links. When this happens, the network then appears to recover from congestion. Thus congestion and alternate routing seem to reinforce each other until measured link qualities fall to values which quench the alternate routing, allowing the network to recover and resume normal, uncongested operation.

> Network performance falls apart when congestion arises. One example of the chain of causes and effects involves the station PR. When it becomes congested, it misses LROPs from what should be good neighbors. After missing three in a row from such a neighbor, it declares the link quality from that neighbor to be zero. This causes apparent disconnection of the station PR from major portions of the net, leading to lack of routes to PRs in those areas.

During the time frame in which these experiments were conducted, the SRI network was suffering from a rash of "bad IDs". These are values appearing in words of the packet header which normally identify PRs on the packet's route. Instead, corrupted values often appeared, looking as if various PRs were part of the network when in fact they did not even exist. This resulted in PDPs to report these apparent new PRs to the station, labeling attempts directed at these apparent PRs, possible packet transport peculiarities, and a tendency toward congestion. Later, the bad IDs were traced to a PR whose cyclic redundancy check hardware was broken. But during these experiments the bad IDs were still plaguing the network, as evidenced by the first of two conclusions from the September 6 run:

> The packet radio network is being severely affected by the bad IDs. Halting transmission of packets destined for bad IDs (by a new PR software release from Collins) has significantly improved network performance.
>
> Alternate routing, even in the absence of severe congestion, increases the network delay. In this experiment, the network was slowed by 100-150 milliseconds (from 250-300 ms with alternate routing disabled by a patch in the PRs, to 400 ms under normal operation.

In the September 4 run we noticed a serious problem in PDP transmissions from PRs experiencing connectivity changes while congested. A buffer was never available for PDP generation. Consequently, the station would not receive a PDP reporting a new good neighbor (the van).

The September 6 run included a change in PR software to call the "scrounge" routine when needed to obtain a buffer for generating a PDP. This routine searches for I/O completions and, if all else fails, will steal a buffer from the transmit queue. Its inclusion has been very helpful in maintaining connectivity information.

The second of four runs on September 18 established benchmark values for traffic throughput and delay in the CAP 5.1.3 mobile network. This recalibration was needed due to the significant performance improvements brought about through the previous two mobile runs. The procedures used were as follows:

1. We used CAP 5.1.3, which incorporates the scrounge routine to provide a buffer for PDP transmission and the bad ID detector which stops transmission of the packet.
2. We used two SPP traffic streams. One originated in the van; the other originated in a TIU located near the station. The stream from the TIU to the van was sent in blocks of 1000 packets. This enabled us to time the interval for correct reception of the entire message. Available statistics included the range of packet delays (from packet transmission to reception of SPP ACK) and the number of SPP retransmissions.
3. PMON traffic from the van bounced off the station at one packet per second.
4. Each SPP stream offered packets as fast as the network would accept them. The maximum possible rate is one full-length packet every 80 msec., roughly 12 packets per second. If the PRs accepted packets at the highest possible rate, this would result in 96 packets per second in the network (two streams at 12 packets per

second, times 2 for the SPP ACK and times 2 for the active ACK to each SPP packet), exclusive of the slight PMON traffic.

The performance seen in this benchmark run was summarized this way:

> This run is to be used as the benchmark of performance for future (and CAP 4.9) comparisons. In this run SPP packets were offered with zero delay between packets; the PR determined the transmission rate using 80 milliseconds as the minimum interval between packets from the 1822 interface. The third set of 1000 packets was sent in an area of difficult connectivity and will be our standard. The total run time was 7 minutes, with an average delay of 350-400 milliseconds (longer than the 200-300 of run 1, due to a route of two hops instead of only one hop). There were three duplicate ACKs received and two SPP retransmissions were required. There were few losses of PMON packets.
>
> And the conclusions we reached from this are:

1. We were not able to congest the network with two SPP traffic streams injecting packets as quickly as possible. Link qualities remain high and control traffic continued to reach the station.
2. The overall performance was good. Packet delays were reasonable and packet loses few. The ability to time the transmission of 1000 SPP packets is a great improvement over "superprogrammer" text printout timings which were subject to variations from host performance.

The first run on September 18 was also very successful; it used traffic offered at 5 full length packets per second on each of the two SPP streams, and found that the network handled the offered rate with little congestion. Packet delays averaged 200-300 milliseconds with peaks of 600 in areas of very poor connectivity. Link quality values remained high.

Runs three and four on September 18 were disasters. SRI tried to install some test patches supplied by Collins, but a misunderstanding led to the patches, designed for CAP 5.1.2, being installed in PRs running CAP 5.1.3. This caused pervasive problems. As one experimenter put it:

> We were unable to open any SPP connections. There was no obvious reason why, but it just didn't work anymore. We decided to flush the first patch and install the other one. We did this on several PRs only to discover that nothing was working properly. The PR's receive enables were going down for long periods and the PRs were faulting in various ways. We tried downline loading the PRs to get the patches out, but even had trouble initializing them. We were getting RAM checksum faults and some other strange things. So we halted all of the PRs in the net, also the station, and started from scratch. The station was rebooted, station PR restarted and rebooted, then each of the other PRs one at a time. PR-21, which had been faulting madly, recovered and was just fine. PR-27, which had been just fine, caught PR-21's illness, and so it went. Three hours later, and much wailing and stomping, the net almost works. PR-27 is running, but if I initialize it, it will fault halt forever. In the van, I had to remove power from the digital section, remove the RAM cards from the case so they would forget everything they ever knew, then reload the PR. As of morning the next day it was still unclear if they would have to send someone to each of the hilltop repeaters to pull their RAM boards.

This problem appears to have been so severe because of a susceptibility the PRs have to garble data in their base page of memory. BBN personnel had commented on this previously (see Quarterly Progress Report No. 18, BBN Report No. 4338, March-May

1979, page 8). At the July 11-12 CAP 5 implementers meeting at SRI, we called for a clarification of what assumptions the PR makes about validity of contents of base page locations. Although it would appear that this is not only a source of frustration and delay when experimenters or users trip over a polluted network, but also a network vulnerability issue, there has not been time to address it further in the PR community. We suggest that such an effort would be time well spent.

The September 28 mobile run was conducted as part of a meeting agenda at SRI, and its procedures were specified by SRI, not BBN. The "superprogrammer" printout was again employed, the station was not attended by monitoring personnel, and various problems arose (outages, the van's PR faulting and restarting, and slow "superprogrammer" printout). The cause of the problems could not be deduced.

One new tool was used this day which was not employed previously: an Interdata-70 and an IPR monitor radio were used as a packet logger to record on tape the entire run. Although faults were later found in the packet logger system, we strongly support its use in future experiments, to provide a relatively complete log of packets on the radio channel. This will permit detailed "post mortem" analysis of experiments, when factors of interest arise which were not even noticed during the run itself.

## 5. FINAL REPORT: EVOLUTION OF ROUTING DESIGN

Over the past 5 years we have transformed the network design for when and where to transmit a packet. The original choice was a highly centralized design in which virtually all user traffic was forced to follow a path into a central node, the station, and then out to a destination. To support the user traffic, control traffic was also forced to flow in this path. The resultant bottleneck was severe and led to user throughput on the order of 1 to 2 packets per second. This protocol was known in the packet radio working group as CAP4 (and preceding). CAP4's basic problem stemmed from how information pertaining to network connectivity was collected; most commonly the status on connectivity between two PRs was evaluated on the basis of one packet (ROP) emitted once a minute and forwarded to the station. This proved inadequate to maintain those links. Consequently, the useful routes were those radial from the station. The radial links were similar to those forced upon the network from the hierarchical routing design of CAP2 and CAP3.

### 5.1 CAP1

The first design packet radio protocol was completed and reviewed in 1975. The review resulted in extensive modifications and additions to the original design which included:

- hierarchical routing except for ROPs
- more ROP information (such as Labeled or not)
- move Label into text so PR can receive same as was sent
- terminal PRs not forwarding traffic
- PRs not unlabeling themselves

## 5.2 CAP2

In 1976 CAP2 was formed out of the initial design and review. The CAP2 network is organized for packet routing into hierarchy levels. Every PR is assigned to a level (the number of PRs in a level is assigned by station operator during initialization) and given a label identification unique within that level. PRs are initially unlabeled and may become so again, if the station is able to transmit an unlabel packet to the PR. This was considered a valuable tool in preventing PRs from unwanted assistance in packet transport.

A PR becomes labeled upon receipt of a label packet containing level, label and route to the station. The route consisted of a label for each level between the PR and the station PR inclusive. User traffic follows the label path into the station and then is forwarded to the destination PR using the outbound label path.

Connectivity information was gathered through the use of Repeater On Packets (ROPs). These packets are transmitted once a minute and forwarded into the station by all PRs receiving the packet directly. The station then evaluates the links between PRs upon the successful forwarding of a ROP from neighboring PR. If a PR is busy, it may delay transmission of ROPs. This added delay is capable of reducing congested network connectivity to nothing.

## 5.3 CAP3

A year's experience with CAP2 pointed to a number of possible improvements. In particular, the labeling requirements to remain within one level proved too restrictive and the requirement of including the entire route in a label was scrapped.

In 1977, CAP3 resulted from the following modifications of CAP2:

- ability to change a PR´s level in the hierarchy to a lower level
- implementation of the station unlabel feature
- incremental routing

Although the change to incremental routing simplified the relabeling procedure, it added complexity to the network as a whole. Knowledge of the route format was not built into the PRs. Rather, part of the station´s role was to inform the PR how to locate the route fields of the packet header that the PR would need to look at. In particular, the PRs must examine fields for their own level and for the next inbound level. While a PR could be relabeled requiring a different format, the overall network level configuration could not change from initialization.

In hierarchical routing, every PR is assigned to a level and given a label unique at that level and a route to the station. Packets traverse PRs at consecutive levels and, if transmission along some link fails, can be alternate routed to any PR at the right level.

As before, all user traffic passed through the forwarder. But CAP3 also included SPP to increase the reliability of end-to-end transmission.

## 5.4 CAP4

The lack of point to point (PTP) routing capability was hurting network performance. Hence, the decision was made to abandon hierarchical routing in 1977.

In the CAP4 protocol, every PR is given a selector (instead of a combination level and label) to identify it in routes and, as in CAP3, a route to the station. PRs may also be given good neighbors for use in alternate routing. If transmission of a packet along some link fails, it can be alternate-routed to any PR that can help it along its route -- i.e., any PR finding itself or any of its neighbors in the remainder of the route.

Later versions of CAP4 replaced the 8 bit selectors with 16 bit PR IDs and included a PTP routing facility.

## 5.5 CAP5

A new routing algorithm was designed to overcome flaws in CAP4. The new design, referred to as CAP5, reduced the volume of control traffic by eliminating the once a minute forwarding to the station of primitive "I'm here and a PR heard me" messages (ROPs). These ROPs were replaced by localized packets (LROPs) which contained significantly more information about each link and were generated every 7.5 seconds so that changes in link quality would be diagnosed sooner.

Under CAP5, the PRs evaluate the local link qualities and only report them to the station if they have changed significantly. Quicker response and more accurate information led to our first effective (PTP) routing. In turn, this increased network throughput by permitting other paths than through the station.

In addition, the Labeler process began to play an important role in network debugging and tuning through expansion of a station operator's user process. This process provides access to important station tables indicating:

- link qualities and neighbors
- best routes between PR pairs
- current Labeling
- time since hearing from PRs
- device/PR correspondence tables

The operators are also able to selectively print packets of interest. The choices range from LABELS, to PTP route assignments and PDP (Performance Data Packet) reasons reported by PRs.

Operators could also prod PRs for more recent information, or to see whether they still responded, by another command which emitted command packets to PRs requesting PDPs.

Momentary network problems led to momentary LABELER printouts. One example of interest was the infamous hunt for the "Bad IDs" in which unusually numbered PRs reported to the station would be printed.

## 6. FINAL REPORT: DEVELOPMENT OF HARDWARE AND SOFTWARE TOOLS

An important aspect of our Packet Radio program has been the development of tools necessary to create the capabilities of Packet Radio.

In early 1975 we chose the hardware and began developing three critical software tools: XNET, ELF and the BCPL library.

### 6.1 XNET

For remote access to the station, we created the XNET (cross-network) debugger. XNET uses a large computer to talk with a small computer across a communications network. The program being debugged, our station, is in the small computer which also runs a compact debugger process to perform examine, deposit, start, stop, etc. commands upon the subject process. The large computer makes use of a large memory and greater processing power to facilitate debugging conveniences. Mnemonic instructions typed to a large computer in Boston can be translated into the simpler instructions for a target machine located in Dallas or Palo Alto or Fort Bragg. The command format was made as similar as possible to that of major existing debuggers such as DDT and RUG.

XNET improvements have been a continuing part of our effort. We added checksums, retransmissions and memory verify commands as well as bug fixes and modifications necessary to keep pace with internet developments.

## 6.2 ELF

Our operating system, ELF, modified from the original ELF developed at SCRL, has proved to be a complex and difficult component. We have expended considerable effort in debugging performance improvements and resource allocation. Eventually, we added disk loading and in-core restart features.

Our current version appears to operate well within its resource allocations and without crashing. The sophisticated process to process communications capability has proved very valuable as the functionality of our station grows more complex.

## 6.3 BCPL Library

BCPL was chosen for implementation of Packet Radio Station software to be run under the operating system ELF. BCPL service routines involving input from and output to peripheral devices attached to the PDP-11 must be modified to access those devices through ELF.

The major modifications to BCPL were as follows:

- o Making use of the "Freeze process" to gracefully terminate execution of user´s program.
- o Allow routines to "CREATE" allocations within ELF for new user processes.
- o Handle ARPA network messages through ELF.
- o Rethinking of BCPL´s control structure for information input to allow for the "Any Input" test. This test merely checks current status; it does not wait for I/O completion and was not directly available in ELF.

These modifications were completed in the first year of our contract and have proved to be efficient.

# APPENDIX A
# BIBLIOGRAPHY OF BBN PACKET RADIO TEMPORARY NOTES

85 Design Suggestions

122 Packet Radio System Design Issues

123 Packet Radio System Capabilities

124 Proposed PRN Protocols

125 Functions and Structure of a Packet Radio Station

126 Point-to-Point Routing in the PRnet

138 Packet Radio Station Hardware Operating System and Applications

141 Cross-Network Debugger User´s Manual

142 Response Time in Cross-Network Debugging

143 Specification of Basic PRN Station Modules

145 Preliminary Cross-Radio Debugger Specifications

147 Modifications to Virtual ELF

156 Gateway Design for Computer Network Interconnection

159 A Proposal for Incremental Routing

162 Routing in the Initial PRNET

165 Will the Real SPP Please Stand Up

174 Packet Radio Network Station Labeling Process

177 SPP Definition

180 Cross-Radio Debugger

182 Packet Radio Information Service

183 Neighbor Table Measurements for Control of the Packet Radio Network

184 Preliminary Functional Specification of the Station Measurement Process

185 Report of Station Software Delivery

191 Terminal-On Packet Proposal

192 Route Assignment Proposal

194 Point-to-Point Routing Proposal

196 Status Information on SPP Connections

199 Some Station Development Issues

212 Specification of Measurement File Entries

215 Measurement File Delivery Specification

216 Specification for an ELF System with Disk and Net Loading Facilities

217 Measurement of Station ROP-Processing Capacity

218 Station Control Module Specification

223 Alternate Routing Reconsidered

232 SPP Retransmission Count Field

235 Proposed Modification to Point-to-Point Routing

238 Transfer Points in Point-to-Point Routing

239 Use of IDs in Routes

240 Use and Abuse of the ARQ Bit in SPP

241 Gateway Dynamic Routing

242 Gateway Routing

245 Symmetrical 1822 Interface Specification

250 Multistation Design Alternatives

255 LROPs and Neighbor Tables

256 Stationless Compatible PRNET Routing

258 Remaining Issues in Stationless Compatible Routing

259 Thoughts Involving LROP Things

260 Specification of a Rudimentary Multistation Capability

261 Resolution of LROP, etc. Issues

264 Changes Necessary for Rudimentary Multistation Capability

265 Issues in Congestion Control: Detection and Current Routing Design

271 SPP Heterostate Diagram

272 SPP Oscillation

276 Specifications of New PR Down Line Loader

277 A Simple Fairness Algorithm

278 New PRN Device ID Policy

279 Internet Routing and the Network Partition Problem

280 Transfer Points

281 Multistation Design Specification