The Racing Problem: A Packet Solution

Robert E. Bruninga
6103 Hillmeade Rd.
Bowie, Md. 20715

Background

Last June several members of AMRAD participated in the Old Dominion horse ride and 100 mile endurance run near Front Royal Virginia by providing mobile and emergency communications. A dozen or so checkpoints were manned by radio amateurs as well as shotgun riders with each of the key event and emergency personnel to provide VHF communication throughout the several county area of rural roads and 1500 foot mountains. A portable repeater was constructed out of two Icom 2AT walkie-talkies and battery powered throughout the weekend event. This was only the beginning of the excitement involved in designing a better way to do it next year, In fact, our primary interest as noted in Dave Borden's AMRAD Newsletter Packet column of July 83 was the desire to link a system of computers on packet radio to handle the data on the over one hundred horses, riders and runners so that information would be readily available at key points for emergency purposes. This paper will describe a distributed data base system implemented with Commodore 64 and VIC-20 Computers linked via amateur packet radio.

Data Reporting

Knowing where all the horses, runners and emergency personnel are located is the single key to rapid response in emergencies. Keeping the time of arrival and status of all race entries at each checkpoint in a large array space would allow easy manipulation of the data to serve a variety of needs. Since series of numbers are already assigned to the 100 and 50 mile horses and runners, a series of numbers assigned to key personnel and emergency services would allow them to be tracked in the same data system throughout the course. A single reporting format can be used for any horse, person, or asset of the form as shown in figure 4.

NUMBER, TIME, LOCATION, DATA

These reports may be used directly to update an N-by-L array of time and status for every unit in the event. Here, N is the number of persons and horses and L is the number of reporting locations. Actually two arrays with the first one recording the time of arrival and the second one containing a coded status or departure time would hold the entire status and historical file of the event.

Data Fields

BY coding the checkpoint names and status, a very short fixed length string may be used for transmitting this data across the data channel. With a three digit number, three digit time, single character checkpoint and status key, this string is only 8 characters long. The format of figure 1 has been designed not only to be as short as possible but also to require a minimum of processing overhead.

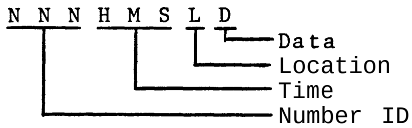

Figure 1. The format of a single race item report consisting of 8 ASCII characters.

The generality of this report is accomplished by assigning numbers to everyone and everything in several compatible number series. For the Old Dominion Ride and Endurance run two such number series have already been used. They are the 100 series and 500 series for the one hundred and fifty mile entries respectively. Using these numbers as a starting point and making some judgments on potential group sizes, the numbering scheme shown in figure 2 is suggested.

l-99 - Ride, Radio and emerg. pers
100's - One hundred mile entries
200's - Runners
300's - Runners
400's - Cavalry
500's - Fifty mile entries

Figure 2. Number series to uniquely identify all race entries and race assets.

Associated with each of these items at any one time is a location. The locations might be gates or checkpoints, or they might take on other meanings as well. Since the data packet going over the radio channel needs to be as short as possible, these location identifiers should be keyed into a single character format. This allows up to 91 keys using the ASC and CHR$ functions over the range 35 to 126. in this way, the time is also compressed into three ASCII characters, one each for hours, minutes and seconds. Some keys for locations along the trail route and elsewhere are shown in figure 3.

ASC	CHECKPOINT	ASC	LOCATION
35 -	4-H camp	55 -	Broadcast msg
36 -	Lands Run	56 -	Front Royal
37 -	Bentonville Br	57 -	Luray
38 -	McCoys Ford	58 -	Detrick
39 -	613 split	59 -	King Crossing
40 -	Yates	60 -	Pala's
41 -	Bixlers Br	61 -	Repeaters
42 -	Woodstock Gap	62 -	Mobile
43 -	50 Finish	63 -	Food
44 -	Edenton Gap	64 -	Gas
45 -	Hickory Lane	65 -	Searching
46 -	Virginias	66 -	Vets
47 -	Seamens	67 -	Unavailable
48 -	Picket Springs	68 -	Gone home
49 -	Shermans Gap		
50 -	613 Split		
51 -	McCoys Ford		
52 -	Lands Run		
53 -	100 Finish		

Figure 3. A table of suggested location identifiers by their ASCII equivalent codes.

The single character keyfields allow some 91 different locations and status indicators to be represented. Several suggested data keys which can serve horses, runners and VIP's alike are as follows:

I - IN	O - OUT
H - HEADED FOR	L - LOST
M - MESSAGE FOR	V - VET NEEDED
S - SCRATCHED	P - PULLED
C - CREW NEEDED	D - DOCTOR NEEDED
F - FURRIER NEEDED	E - OUT TO LUNCH
ETC	

Packet Radio Data Distribution

Recognizing that the most important real time data on any item in the data base is only the last reported status, the organization of the data network as a distributed system could make it much more survivable in an amateur/portable environment. Also the KISS principle (Keep it Simple, Stupid) could be followed more closely. Under this concept, there is no central computer and all the field display computers need to maintain only the last known status on any single item in the data base. To minimize data channel load yet provide for full refresh of the data at the display terminals, a scheduler in each computer would periodically retransmit all data entries for which it was responsible. Because of the one-to-many distribution of the updates to all other display terminals, the packets will be transmitted and received in the packet monitor mode error free, but without acknowledgment. Sending one of these refresh packets every number of seconds would assure complete data base refresh every few minutes or so. With 8 characters per data item, a concatonated combination of eight such items per line would make a nice size packet as shown in figure 4.

```
#uTNNNHMSLDNNNHMSLD......NNNHMSLD
 ||  item #1  item #2      item #8
 |└Transmitting station identifier
 └─Update identifier
```

Figure 4. A single packet frame consisting of eight individual item reports.

This single string format allows simple Basic input commands to be used to input the complete string at the display terminals in one operation and minimize packet overhead. The line can be verified prior to array update by testing for the update identifier and fixed length of 67 characters. If a display terminal crashed or lost its data, it could be allowed to slowly rebuild its data or it could be completely updated by any other computer in less than 30 seconds on a dedicated basis at the 1200 baud channel capacity.

Display Processing

Once received, the updates are entered into a string array of the form S$(N)=HMSLD where N is the item number (which may have to be hashed for effecient use of array space if multiple item number series are used) and HMS, L and D are time, location and data keys respectively. The individual display terminals then can be programmed in a variety of ways to provide querie/response information on the latest status. Several possible screen display formats such as the following could be implemented:

* locations of VIP's and Emergency pers
* Top 20 Horses
* Last 20 Horses
* Horses at location X
* Status of Horse Y
* Horses departed location X
* Missing horses
* ETC

The possibilities are endless and free to the imagination of the display terminal programmer. This flexibility is necessary due to the variety of computers and screen size formats which will be used.

Notice that nothing prevents a single computer from building the complete set of N-by-L arrays since all of the data will have been transmitted on the channel. In fact, 16K or larger and Disk based systems should be programmed with this capability. Also note that as checkpoints are completed and all race entries have passed through a particular point, there is no further requirement for packet refresh from that station. That packet station can then be moved to a later checkpoint for reuse.

Data Channel Activity

As the flow of **the** race progresses, data channel activity will tend to **migrate** from station to station as shown in figure 5. For a race of **160** entries, the starting point station will initially be responsible for all 160 reports, which, in groups of 8 corresponds to 20 packets. If one packet is transmitted every 12 seconds, a complete update is provided every 4 minutes. This cycle will continue until the lead horse arrives at gate two. As reports are initiated from that station, station one at the starting point will see the reported location field change and stop refreshing that entry on a one-for-one basis with reports it hears from the new reporting station. This is very simple since each station uses the location field to identify its own reports which it is responsible for updating. In this manner it can be seen that there will always be a nominal 20 packets per period being transmitted and that the peak load will move as a bell curve distribution from station to station as the race progresses. There will also be a nominal background level of packets from other stations reporting the movement of VIP'S and other status throughout the course.

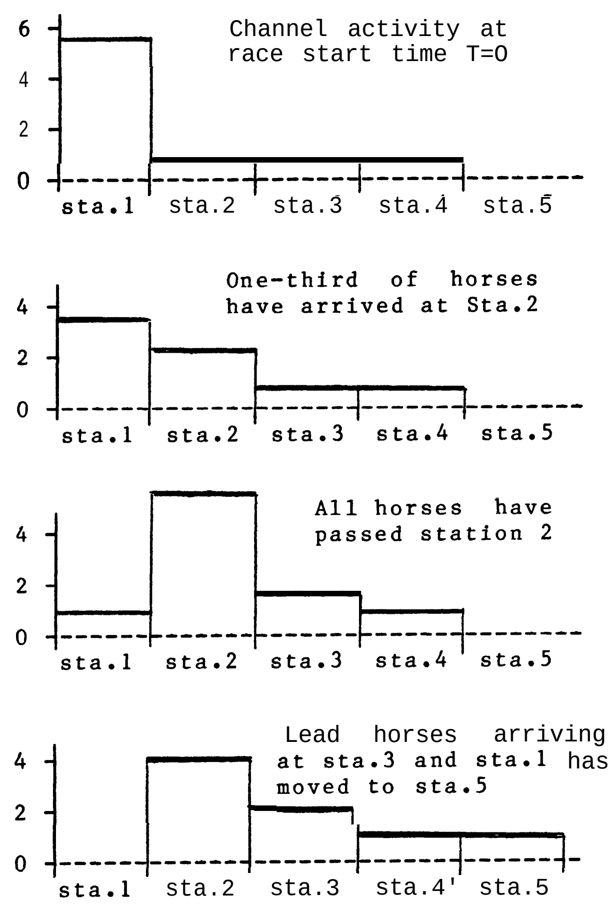

Figure 5. Channel activity in packets per minute remains relatively constant although reporting responsibility moves from station to station.

Station-to-Station Messages

An enhancement **to the basic database** system described above is the **addition of a** message packet format **that allows the** exchange of text messages among the packet stations. The format of **figure 6 is used.**

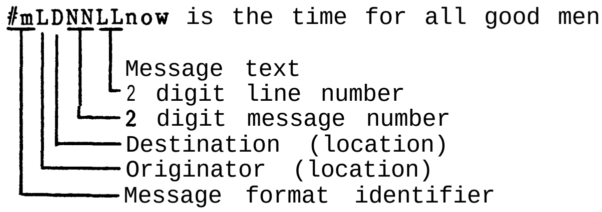

Figure 6. Message packet format includes line number and message number **to assure message** integrity over the unconnected net.

Point to point messages are programmed to require a specific acknowledgement, while system-wide messages or broadcasts are scheduled for periodic update. Smaller systems (4K) may retain only the last message while larger systems may desire to retain all messages until cancelled.

Data Array Compression

For **array** processing, the individual display terminals should be programmed to accept each **report** and use the time, location and status to update its **data base.** To save memory, the gaps in the item number series suggest the computation of offset values for each of the number series and using these offsets to compute an actual array address. This means that on initialization, the program queries the user for the total number of entries expected in each of the number groups and uses these as offsets for the array subscripts. Using this approach, the total array size needs to be dimensioned no larger than the sum of these offsets and only a single computation needs to be performed for each array access.

The scheme to significantly compress the number series into a contiguous array is as follows:

```
INPUT"Number of  series ?"; NS
FOR I=1 TO NS:
  INPUT"Start and end values?'; S(i),E(i)
  R(i)=E(i)-S(i)       :RANGE
  A(i)=A(i-1)+R(i-1)   :ARRAY OFFSET
  D(i)=A(i)-S(i)       :DELTA OFFSET
NEXT
```

Now, given a horse number H, its **array** location, A, may-be found using the loop:

```
FOR i=1 TO NS
IF H<E(i) THEN A=H-D(i): i=NS
NEXT i
```

And any array number, A, may be converted back to an item or horse number using the loop:

```
FOR i=NS TO 1 STEP -1
IF A>A(i) THEN H=A+D(i): i=1
NEXT i
```

Finally, some consideration should be given to the scheduling of packet transmissions according to the loading of each particular station. Using 4 minutes as a nominal refresh cycle period, each station should time the delay between each packet transmission inversely proportional to the number of packets it needs to send or a minimum of once per minute if he only has one packet. With a peak load of 200 reports or 25 packets per period, the minimum delay between packets should be about 10 seconds resulting in the following relation:

$$N = INT(R/8)+1 \qquad D = 290/(N+4)$$

Where D is the delay in seconds between each packet transmission, N is the number of packets due for transmission computed from the number of reports R.

The purpose of this paper is to suggest early agreement on the format of the data distribution packets so that AMRAD packet owners can begin working on the display formats and querie/response capabilities of their individual systems. They may add as many bells and whistles as they feel necessary, such as pointers to string arrays containing the full names and statistics of all horses, runners and people. Dependjing on these bells and whistles, there should be no problem fitting up to 200 horses, runners and VIP's into less than 4K for a VIC-20.

The requirement to provide emergency communications coverage for large public race events is a frequently repeated amateur radio public service event. Hopefully bringing the benefits of packet radio to bear on this application will demonstrate the tremendous potential for this state-of-the-art mode of communications.