

Packet Radio Logs The OGAR!

Jim Baremore, K5QQ
k5qq@arrl.net

What is an OGAR? How do you log it? What's this all about? Whew!

Easy now, slow down. OGAR is the annual Ozark Greenways Adventure Race. The OGAR is a grueling, but fun, 8-14 hour race composed of four-person co-ed teams who bike, run, canoe, orienteer and perform certain mystery events in a very competitive environment. Held in the middle of Missouri in early June, it is also hot and dusty! This year a new feature was added: Packet radio was used to log contestants as they proceeded through the eight checkpoints along the course. Computer programs written specifically for this application allowed race officials and spectators to view the real-time progress of the teams on the course.

This year's event was the sixth, and it has been growing in popularity every year. There were 392 contestants who paid the \$100 entry fee to compete. Funds raised by this event are used to support walking and bike trails in the Springfield, Missouri area. All of the support for the race is provided by volunteers. A few corporate contributors cover the costs of food and other race supplies.

Hams Provide 2 Meter Communications (The Only Communications)

The first race was held deep in the Missouri Ozarks and in that sparsely populated area, away from the major highways and cities, there is zero cell phone coverage. The race organizers tried using CB and FRS radios to communicate across three different counties and the very rugged terrain found in the Missouri Ozarks. It did not take them long to figure out they needed help. That is where the Christian County ARES Emergency Coordinator, Pat Conway, WA6JGM, and Ken Baremore, W0KRB, the Greene County ARES EC stepped in to help. The two ARES groups provided 2-meter voice communications with ARES members being deployed at each checkpoint and other critical race junctions. They passed on reports of where teams were dropping out and needing to be trucked back to the finish line. Their ability to communicate became essential to help insure the safety of the participants as well as the overall coordination of the race.



K5QQ

7:01 AM and the race is on. The first racers can be seen on the road while others are queued up to be on their way.

The race data, e.g., team times and status as they went through each checkpoint, was recorded by the race official at the checkpoint using a pencil and a clipboard. It was used to validate the winning team's performance and was also an attempt to keep track of the teams to make sure none were lost on the course. A lot of radio chatter was generated when discrepancies were found in the reports! After the race, the data from each checkpoint was manually transcribed by the Ozark Greenways staff and displayed on the Ozark Greenways Web site. That process typically took a week or so.

Finding a Team's Status was like Finding a Needle in a Haystack

Since the race lasts an average of ten hours, there were many family members and other supporters who would show up well after the race had started. They were interested in where their team was and how they were doing. Without any information on how the race was progressing and specifically how their team was doing, they could only try to drive around the course asking at each checkpoint if their team had been to that location yet and where they might go to see them in action. Of course, that information was seldom available and with the small country back roads the race took place on, this vehicle traffic going to each of the checkpoints was detrimental to the safety of the very racers they were interested in.

Although the ham radio voice communications assured no remote sites were isolated

from contact with the HQ, they offered no practical way to provide real-time race information. Packet radio operating in a terminal mode offered nothing to the race as there were no lists or records to be communicated. Likewise, merely sending Packet reports that contained the same data as on the written sheets would only be a backup for the hand-written sheets and offered no time advantage or enhancement to the race. The results still needed to be transcribed into a spreadsheet in order to calculate lap times or to determine who was in what position during each leg of the race, etc. In other words, all of the things racers and their families are interested in!

A Real Opportunity for Hams to "Raise The Bar" of Support

What the race needed was real-time results at the finish line and a data entry process that was as quick as merely using a pencil to check a team off a list and note their time. To visualize the problem, the first checkpoint in this year's race logged a team every 18 seconds, a logging rate that is equivalent to working and logging over 200 QSOs per hour. That's way ahead of my recent Field Day rate I assure you! Obviously, everything had to be optimized for speed and efficiency. In spite of the tough problem, the advantage of real-time results was also impressive. Not only would it assist the spectators, it could also show problems that might exist on a leg of the course. It would provide immediate feedback on teams that had become disoriented and missed a checkpoint. It could also assist in the efficient

utilization of volunteers deploying the various logistical equipment. It could clearly add to the safety of the participants.

Last fall, I decided that maybe I should try to write a program that would provide this needed information. I had written several other programs for my SAR team in New Mexico and now that I was in Missouri, thought I would try to make a contribution here. The concept was certainly easy enough so I started working on the program, thinking it would only take me a month or so to do. I also thought it would be a great showcase for spectators to see that ham operators had a lot to offer to the community in addition to the normal voice communications they had come to expect. I was pretty excited about how this was going to work out.

My little project had evolved into six custom software programs integrated together to support the race needs. It seemed I was working on my computer day and night getting everything figured out. I had to learn how to do *Windows Sockets* for the client server applications and how to do the serial port communications so I could talk to the TNCs. It was a lot more than I had expected but it was "looking good."

The programs allowed teams to be quickly checked in at each of the eight checkpoints along the course. Then, with packet radio providing the data transfer between the remote sites and the HQ station running other custom programs, that data was immediately processed for validity and a QSL report returned to the originating station. The validated data was then sent over a Local Area Network to yet other programs which displayed the results on an *Excel* spreadsheet at the finish line. Race officials and family members were now able to watch the race progress in real-time.

Midway through this project, most everything was functional although not working without problems. I gave a demo to my brother Ken, W0KRB, and he immediately called the Ozark Greenways folks and insisted they come and see this program and what it would do. A week later a demonstration was given to several of the race organizers. It was an instant hit and ham radio and our ability to log the race with real-time results was about to be on center stage. Now the heat was really on to make this work. Nothing like a little pressure to keep the midnight oil burning! Although it took a lot longer than I expected to get some of the issues resolved, I just knew this was a great application for the old packet radio equipment several of us had sitting around in our radio rooms.

Race Reporting Uniquely Suited for Packet Radio

There are applications in which a packet radio link is configured to support Internet



Jim, K5QQ, at the start of the race. The computer is inside the box to make the display move visible in the sunlight.



This screen shot of the *Data Linker* program shows the third location, the Canoe Leg, at the end of the race. All of the 91 possible teams have gone through the location with all but one of the teams complete.

outages or other situations where an Internet connection is not available. Those applications typically support e-mail clients or other office like applications. The *WinLink 2000* program discussed in a *QST* article is one such application. Another recent *QST* article discussed how to modify commercial wireless LAN equipment to provide LAN connectivity to a remote location. However, the OGAR situation did not involve conventional applications looking for an Internet connection to link them together. There were no existing programs which would easily collect the data nor were there programs to process such data and display it in the required fashion. Filling out a spreadsheet at each remote location

and sending the sheets back and forth was neither appropriate to the task nor a viable solution. This was a unique situation as time was of the essence and the amount of data to be transmitted was quite small. It was a great way to use the unique characteristics of packet radio for this real-time data transmission, but it required some custom software to be written to make it work.

I designed the software so packet radio was used as a data transmission media, much like a telemetry link, instead of a keyboard-to-keyboard link or a file transfer mechanism. Once connected, all transmission between stations was under control of the software program. The ham operator merely monitored the link to be sure the path was good and that the hardware was functioning normally. All data was gathered by the program and sent automatically to the HQ station.

This greatly simplified the training and knowledge the operators needed to use the system. That was the secret to the outstanding service the ARES hams were able to provide to the race this year. The ARES members who used the program were quick to comprehend how to use it and the three training sessions that had been set up were used mostly to check out the hardware and be sure everything was working great.

Data Linker Software was the Workhorse at the Remote Locations

One of the major programs in the set of six is *Data Linker*. This program is used at the remote sites and makes it easy for the operator to quickly enter a team number, tab to one of the status indicators and then send a time stamped report. The data report from the *Data Linker* program was sent via the packet radio link to the *Data Display* program which ran on the HQ station computer and collected all of the remote reports.

The *Data Linker* program was designed to allow the operators to connect directly to HQ or to go through a digipeater with just the touch of a button on the software display. Information about the status of the contestants who had passed through their checkpoint was displayed in a separate window. The program also included a terminal window to allow the operator to monitor the TNC traffic. Custom scripting of the TNC commands was controlled by the program to insure critical settings were preset. Packet timing parameters were also set to allow efficient use of the portable digipeater which was deployed during the event for those sites which would have to rely on the digipeater to get their traffic back to the HQ site.

Some of the remote locations had two co-located check-in points for the teams. Separate computer logging stations were

deemed necessary to insure teams would not accidentally check into the wrong station, ultimately leading to their disqualification for missing one of the race legs. Rather than forcing each logging station to have its own packet radio system and the likely interference from the co-located operations, I wrote another program, *TNC2LAN*, as part of this evolving suite of programs. It allowed two reporting computers to link together over a LAN connection and share a common TNC and radio.

Each location's reports were generated separately by the two versions of *Data Linker* that were running and, although sent over a common TNC and radio, were treated like they were two autonomous systems. This software not only reduced the hardware requirements almost in half, it also required no change in the operator's procedures if the location was sharing a TNC and radio or operating in a stand alone mode.

Finish Line Only Slightly Less Complicated

The contestants were last logged as they

crossed a finish line position which was only about 100 feet from the HQ tent. Like the other locations, the *Data Linker* program operating from that finish line could connect via packet radio. But with the short distance between the two positions, I also included the ability for that location to connect via a wireless LAN. This offered a lot of contingency for unexpected problems—and they did happen!

At the HQ tent, the *Data Display* program processed the data and did a data integrity check. It then sent a QSL report back to the *Data Linker* program confirming receipt of the report. That QSL report also allowed each site to keep track of teams that had dropped out of the race prior to their location. That kept them instantaneously informed of changes in how many teams they should be expecting to go through their checkpoint.

The *Data Display* program also sent a file to the computer running the *Excel* spreadsheet. A Microsoft Visual Basic for Applications was the third major piece of code and did all the spreadsheet manipulations. This VBA code checked all of the data to watch

for teams that had missed a leg and would wind up being disqualified. Sometimes they would continue to run in the race checking in at subsequent checkpoints. Teams may have a member drop out and decide to later join back up. The software watched for that event also. In case a team was incorrectly logged by a remote location, the VBA code allowed manual overrides to be entered on this HQ program. That provided a good audit trail of any such corrections. Needless to say, this program processed a multitude of scenarios before the results would be sorted and the winners of each leg, as well as the overall winner, were certified.

Due to the sequential nature of this kind of race, the overall need for computers was fortunately reduced as the hams set up a station at the first reporting locations and when the teams had all moved through that location, quickly moved and set up operations at a more distant station. A lot was riding on their ability to quickly move and redeploy their assets. Nothing like a challenge to hams to get them moving!

[To be continued next month.]

Packet Radio Logs the OGAR—Part 2

Jim Baremore, K5QQ

Continued from January QST, pp 80-82.

Packet Radio did not Eliminate the Need for Net Control!

The race course geometry has always required a net control operation where the net control was at a remote location separate from the HQ. Voice traffic was directed through the net control on a 2 meter simplex frequency. To assist the net controller, another program was written as part of the suite of programs written to support the race. *Data Monitor* was a passive listen-only program which decoded the packet traffic and presented net control with a real time display of how many teams had passed through each checkpoint and how many were due to pass through. This helped him keep a very close watch on the progress of the race and allowed him to redeploy resources in the most efficient manner. Pat Conway, WA6JGM, is the ARES Coordinator for Christian County and manned the Net Control station. He sat from dawn to dusk at a small pullout on the side of the road performing that chore as well as overseeing the digipeater which was collocated at the high spot of ground.

A Good Measure of Complexity is how Simple it Appears!

Although the logging concept is very simple to comprehend, the programs took over six months to write and test for almost any scenario. Three training sessions were set up for the ARES members who would have to set up their computers and networks in the field. It was almost like getting ready for a successful field day operation.

The three major programs worked asynchronously in the event any one part would fail or lose connectivity during the race. This allowed the system to come back up on line and recover data without manual intervention. If a remote site lost their packet connection, the reports would merely continue to accumulate and the QSL status flag would indicate reports were not going through. Timers prevented a reconnect from flooding the packet channel with reports. The Excel spreadsheet VBA code was also operated asynchronously so the data transfer could be stopped while interim results were printed out. There was also additional error checking done on the transmitted data. A CRC string was internally appended to



Food line at headquarters showing the beam antenna used for the packet data link and the hungry participants after the race.

the data for each data set sent to insure data scrambles would not cause spurious results. This process, along with the internal CRC check of the packet radio system itself insured the data would be accurately received.

The system worked flawlessly. Family members were able to check in at the HQ location and see how their team was doing, how they compared to the rest of the racers and what a likely time for them to arrive at their next checkpoint was. There was a constant buzz at the HQ tent watching the race progress and comparing the lap times.

Old Chinese Curse: “May You Live in Exciting Times”

This success did not happen without some exciting moments however. Because of the need for a printer and PA systems at the finish line, the headquarters computers did not have to operate from battery power like the remote stations did. AC power was routed from an outlet by a well pump some 200 feet away. About midway through the race, the printer was turned on to get ready and print out results. It was a laser jet and they draw a tremendous amount of current. It turns out an RV had plugged into the same outlet on the pole and the combination of the printer and the RV’s air conditioner was enough to pop the

circuit breaker. That went unnoticed until the main computer, running both the *Data Display* software as well as the Excel spreadsheet, had the battery go low and the computer suddenly went into hibernation. It was not immediately obvious what had happened and since the computer was only two days old, the worst case scenarios were racing through my mind! When power was restored, everything came back up and the asynchronous design built into the interlinked programs allowed operation to continue as if nothing happened! The many hours of planning paid off.

Unfortunately, one of our members had a medical emergency and had to withdraw the night before the race started. Fortunately all of our advance training and preparations paid off as when the race started in the morning, equipment was in place for that station and operators were ready.

The need for extensive testing and contingency planning was due to the very large race being supported and the spotlight that was on the ham community to demonstrate the system could meet the rigors of a field deployment. A success here would go a long way to adding credibility to the ham community as being one of responsible people helping their community. Good PR is always helpful for hams as we face increasing battles in our pursuit of the hobby. The complete software system, as well as the hams that made it happen, received rave reviews from the race organizers and officials as well as the team members and family who saw the results. We made a very positive impact with this year’s performance.

Overwhelming Logistics

It may not strike the casual reader about how large and important this race was. If the fact there were 392 contestants from nine different states doesn’t get your attention, think about these statistics:

There were over 1600 hours of volunteer time expended on race day alone. Five paramedics and one emergency room physician administered 18 bags of IV solution to dehydrated racers. This dehydration occurred even after 620 gallons of water were distributed by the water trucks. 22 cases of bottled water, 16 cases of soda, 18 cases of Gatorade and 3.5 kegs of beer were consumed at HQ. They also drank 4.5 gallons of coffee at the racers briefing at 5:30 AM! It’s hot and dusty in Missouri this time of the year.

Although this would not have been possible without the software programs especially written for the race, just as important to the success was the hardware required for the eight remote locations. Each one had to have a voice radio for communications of race information and general race control. Each location also had to have at least one packet station and two computers interconnected via a LAN with the attendant routers and cables. Batteries had to support the equipment for the duration of the race. Backup equipment to contend with any emergency was a part of every locations inventory. These locations were deployed along the river banks for the canoe portions of the race, in the middle of a state conservation area for the orienteering part of the race, and at a shooting range parking lot where there were several simultaneous events.

The finish line and HQ operations with all of the people milling around tripping over cables and poking on the computers was another set of unique problems. Of course the biggest problem encountered at the finish line was the grill running just a few feet away. Several of us gained many pounds just smelling the food!

The hams that brought the computers and LAN hardware to the field and manned

Report	QSL	Team	Status	Time
78	Yes	65	C	2:44:52 AM
79	Yes	61	C	2:45:04 AM
80	Yes	16	C	2:49:09 AM
81	Yes	81	C	2:11:00 AM
82	Yes	82	C	2:56:08 AM
83	Yes	30	P	2:56:17 AM
84	Yes	91	C	2:57:41 AM

This screen shot of the Team Report Status is called the Data Linker Program and shows the partial team was team number 30. This display lets the operator scroll back through the reports to resolve any questions which occur at the location. The Ozark Greenways maintains a Web site at www.ozarkgreenways.org, and the Greene County ARES Web site is at www.qsl.net/gcares/.

the computer logging stations were an enthusiastic bunch and were dedicated from the start to making the computer programs successful for this year's race. They included the ARES coordinators from Christian County; Pat Conway, WA6JGM, and from

Greene County; Ken Baremore, W0KRB. Members from both organizations included Randy Atkinson, KA0IQM; Sue Baremore, W0SUE; Barb Higinbotham, WB0YDV; Dick Higinbotham, K0GL; Travis Scudder, KC0AKL; Frank Stogsdill, KA0JIQ; Jean Thomas, K0JAT, and myself. Terry Shoemaker, KE4LQW, was ready with his equipment but an unfortunate last minute medical problem prevented him from attending.

All totaled there were 18 hams involved in supporting the communications needs of the race. The Ozark Greenways organization held a picnic a week after the race to thank all the race volunteers and made a special presentation to the ham community for their outstanding support and very positive support they gave to the race. We're already looking forward to next year's race!

Jim Baremore has been a ham since December 1955 and has earned a BS, MS and PhD in Electrical Engineering. He started writing software programs about four years after he retired, and all of his efforts have been directed to supporting SAR and ARES activities. His primary interest is CW, although he spends a lot of time programming. Jim may be contacted at k5qq@arrl.net.