

From the Editor

I've been lucky enough to work from home since 1998, but for many years I commuted in the Washington, DC, area. Every morning and evening I would blab on a 2-meter repeater that was frequented by many Potomac Valley Radio Club members. One of my favorite QSO partners was Jim Headrick, W3CPB (later W3CP and an SK in 2011), who was first licensed at age 15 in 1932 and did award-winning work on radar and digital systems for decades at the Naval Research Laboratory.

Jim worked early on with Claude Shannon, who changed the standard thinking about bandwidth from purely theoretical to a very relevant focus on information transfer over real-world noisy networks. The Shannon capacity theorem defines the maximum amount of information that can be successfully transferred over any channel. Successful information transfer requires a "reduction in randomness" — the receiver must learn something it didn't know before, a very important aspect that also implies quality/correctness of what is received.

The roots of contesting come from a desire to prove amateur radio can be useful in sending messages without wires to provide reliable communications during emergencies. The first amateur radio contests were very much focused on the information transfer aspect of communications. The ARRL "International Relay Party" and "January Contest" (later

known as the ARRL DX CW and ARRL CW Sweepstakes, respectively), announced in 1929/1930, stressed accurate delivery of fairly complex messages. For years in Sweepstakes, successful two-way exchange of a minimum ten-word message (including call signs) was required for each station to gain two points.

Modern contesting has added many out-of-band tools that work to reduce randomness without much operator intervention: history files, logging software autofill and partial checking, skimmers/spotting, etc. And the exchanges in many contests are pretty well known in advance. But accuracy is still important — if for nothing else to stay true to contesting's roots, but also to maximize your placement in contest results. When everyone has access to the same tools/technologies, a 1.5% error rate in CQ WW CW versus a 4.4% error rate (on opposite sides of the 2.9% average) may move you up several places in the standings.

This is all a long-winded way of saying check out Tree's, N6TR, new "Golden Log" record in the February CW Sprint — 371 Qs. That was also the highest number of QSOs made in this running, though first place Andy, N2NT, was only 5 Qs behind and took the win with 1 more mult than Tree.

A Golden Log in any contest with a serial number in the exchange is impressive. Getting everything right in the chaos of a Sprint is awesome.

Teaching me how to do woodworking, my dad never said a word, but I remember helping him put in crown molding at my grandfather's house. At first, I thought I was much faster than my father; then I realized his "measure twice, cut once" approach was way faster (and less wasteful) than my "measure once, cut 5 times" way of working. Not to mention that his corners didn't need any caulking!

In contesting, accuracy is about getting the call and exchange right, but also making sure the other person gets our info right or even works us at all. The next wave of testers will likely be coming from FT8 as their starting point, where call sign and payload accuracy are almost guaranteed by error-correcting encoding — but we've already seen how the "RR73/final 73 or not" confusion leads to NILs! Not to mention how forgetting to send in your log to the sponsor after posting to 3830 Scores really hurts your score.

Every wave of technology (memory keyers, computer logging, packet spotting/skimmers, etc.) seems to reduce the skill needed to rack up points in a contest, but the cream still keeps rising to the top. Computers have also enabled much better log checking, and new ways of using them to detect cheating aren't far away. Accuracy in "reducing randomness" should always be a top goal in contesting. Congratulations again to Tree for that telephone book-sized Golden Log!

