

A Short Introduction to the Repeater Audio Archive Projectⁱ

by Richard Hagemeyer, VE3UNW

This project was inspired by bad audio. We've all heard some over the air. Almost always, we know exactly what is wrong and exactly how to fix it, at least, that's what we convince ourselves to believe.

I heard some bad audio during the last Canadian Ski Marathon. I was (fortunately) not in a position to break in and offer my wisdom, but I did have a computer and an Internet connection. That's it! I would record some offending audio and email it to the station. Then they could hear what I am hearing and take corrective action.

What a wonderful idea! Well, that was about three months and about eighty hours of effort ago.

How the system was developed from the beginning or how it changed along the way is no longer important. What it looks like now and where it might develop is the subject at hand.

Although this is a radio project, most of the “heavy lifting” is done by software on a couple of computers running Linux. I followed a UNIX approach to the design which basically means that a series of smaller programs will be used and each will do some work and pass the data on to the next program. As well, a client/server approach was used; this is where one computer acts as a client and feeds data (audio) to another computer acting as a server. The client and server do not have to be separate, but certain design issues are more easily solved by assuming they are different machines running in different places.

The program which forms the foundation for the entire project is called *listener*ⁱⁱ. This program will monitor a sound source for audio and record when audio is present. It has numerous other features, so if you run *NIX (this means any variant of Unix or Linux), check it out.

As seen in the data flow diagrams, *listener* is only used at the beginning of the process. The recorded sound then passes through *SoX*ⁱⁱⁱ. *SoX* is known as the “the swiss army knife of sound processing programs” and can do useful things like convert sound formats, remove silence and do additional filtering. A small script gets this done and prepares the file for transmission to the host computer.

At this point we have a smaller, filtered, compressed mp3 file to send via *rsync*^{iv}. The *rsync* client communicates with the *rsync* server and places the file in the correct directory on the host. The pre-process feature of *rsync*, allows for the creation of the directory if it doesn't exist and the post-process feature allows a *PHP*^v program to properly add the data describing the sound file to a *postgresq*^{vi} database.

This is the entire process from hearing the audio on air to making the audio available to a web server for anyone to browse.

All the web pages on the RAAP website are dynamically created by *PHP* which reads the *postgresql* database to decide what information needs to be displayed. This means the latest information is available to the user. The user navigates, from a station selection to a year, month^{vii}, day and finally an hour selection. Within the hour selected, all the sound files available are presented to the user. The page also shows the user the duration and file size for each available sound file.

Next Steps

There are a number of improvements I'd like to pursue. The hardware front-end needs some work. I currently use a dual-band radio capable of V-V operation. I believe this contributes to cross-talk, causing the *listener* program to kick in when it shouldn't (these mostly get eliminated by software). Using separate and cheaper radios should fix this problem. I'd also like to use some form of AVC (Automatic Volume Control) to adjust the audio levels to what the sound card expects. Again this issue is currently mitigated using software AVC. The hardware AVC circuit can also provide some electrical isolation between the radios and the computer, probably a good idea.

On the software side, I'd like to try using a second (or even third and fourth) audio card to accept audio from more than two sources. Some research in this area suggests that four sound cards would work fine. The client-server architecture allows for multiple clients to connect and store audio in a central location.

This raises interesting possibilities of a "clearing house" for amateur repeater audio. Twenty-four hours of audio take about 300Mb of space. VE2CRA has about twenty hours of audio a month and VE3OCE has about ten hours of audio. I think other local repeaters (VE3TWO and VE3RIX) see more "action". Assuming a busy repeater at 1Gb per month, that is only 12Gb per year. The current price^{viii} for a hard drive is \$158 for a 500Gb drive. Even allowing for disk mirroring (RAID 1), this means a years worth of data would cost \$8.75! 500Gb would hold over forty years of data! The hard drives would surely fail before they filled. On the other hand, if forty repeaters stored their audio in this fashion, a year's worth of live audio data could be stored on a 500Gb raid array for under \$10 a repeater.

The power of technology is progressing, it's time to join in.

I wish to acknowledge and thank the Monday night Montana's^{ix} gang for their ideas and encouragement. Without them, I would not have entered the project in the OARC^x home brew contest.

- i **Repeater Audio Archive Project** by Richard Hagemeyer, VE3UNW (<http://raap.admin2.ca>)
- ii **listner** by Folkert van Heusden (<http://www.vanheusden.com/listener/>)
- iii **SoX**, originally by Lance Norskog and now by Chris Bagwell (<http://sox.sourceforge.net/>)
- iv **rsync** by Andrew Tridgell (<http://rsync.samba.org/>)
- v **PHP: Hypertext Preprocessor** originally created by Rasmus Lerdorf and now many people (<http://www.php.net/>)
- vi **PostgreSQL** originally by Michael Stonebraker and now by many people (<http://www.postgresql.org/>)
- vii Original month calender by Dr Quincy (<http://www.drquincy.com/resources/tutorials/webserver/phpcalendar/>)
- viii Seagate 500GB Barracuda 7200.10 SATAII - HD-ST3500630AS, May 13, 2007 (<http://www.pccyber.com>)
- ix Montana's Cookhouse, 1711 Merivale Road. All are invited every Monday night from about 20:30-22:00
- x Ottawa Amateur Radio Club (www.oarc.net)