A Proposal for a Cloud-based System for Testing Communications Designs

Aaron B. Wagner
School of Electrical and Computer Engineering
Cornell University
Sept. 30, 2017

Courses on digital communications taught by IT Society members are often quite theoretical. The "standard" digital communications course, versions of which I took as both an undergraduate and graduate student, abstracts away many of the important but untidy aspects of the problem, such as synchronization and channel uncertainty. As a consequence, while students who complete such a course are very comfortable thinking about digital communications at a high level of abstraction and performing certain kinds of mathematical calculations, they are unable to build working communication systems. Such courses are not very appealing today's students, which leads to low enrollments and a relatively small number of students pursuing a career in areas related to the Society.

It is preferable to instead offer a course that blends rigorous theory with hands-on design. While adding a software-based design module to an otherwise-theoretical course provides significant value to the students, it does not compare with a design project involving a real-world channel (defined amusingly by a colleague of mine as "a channel for which Murphy's law has an opportunity to kick in") with all of the messiness, frustration, and joy that it brings.

The are arguably two impediments to establishing such a course. The first is that resources may not be available to provide the dedicated space, specialized equipment, and supervisory personnel that such a class requires. The second is that many instructors of conventional digital communication courses are unable to commit the time required to set up the necessary equipment.

The Commcloud Option

Both of these impediments could be addressed by creating a "channel in the cloud" service that anyone could use. This would consist of several physical channels connected to one or more servers that are accessible via the Internet. We envision that some of the channels would be audio band (i.e., baseband), to allow one to communicate over a real channel without the complexity of high-frequency carriers. Some of the channels would be radio frequency (RF). Some of the RF channels would communicate over wired channels so that one could test systems at various frequencies without regard to regulatory constraints. Others would communicate over actual antennas, using unlicensed bands, amateur radio bands, or perhaps over a licensed band under a special US FCC exemption (which has been obtain in the past by my institution). Another channel could be underwater acoustic, possibly using a tank directly connected to the system or possibly connecting seamlessly to existing channels at sea. I have been in contact with a group at MITRE Corp. which has communications hardware deployed in Rhode Island Sound that could be used for this purpose.

One of the salient features of this system would be a slight learning curve for the users. A user would only need to create a WAV file of the baseband samples that she wishes to send over the channel. She would then send this WAV file to the server using either a command-line script or a web interface, specifying which channel it should be played over. The server would create a recording of what the receiver heard during the transmission and send this to the user, again in WAV format. The server would queue up requests and serve them first-come-first-served if requests for a particular channel are received while the channel is in use. In particular, the user need not be familiar with GNU radio in order to use the system. This is in contrast to other Internet-connected radios I have found.

Such a system would eliminate most of the impediments to including design in digital communication courses. By using this “commcloud” service, a school could offer a design-based course without a local infrastructure. This would allow under-resourced schools to offer a digital communications course with a substantial design component. Moreover, it would not require local personnel at the school to debug radio hardware; they would only need to master the WAV interface. It could also be used by student project teams who wish to test their designs on real hardware without the overhead of procuring and debugging radios. It could be used by hobbyists, and by prospective Ph.D. students who wish to demonstrate their proficiency in digital communications to potential graduate schools. All of this would facilitate education in digital communications and hopefully draw more people into the field.

The State of the Art

Cornell currently has a functional, but primitive, commcloud system. This system consists of a single Mac Mini server with several external USB soundcards. These soundcards are used as transmitters and receivers for audio-band channels. The channels themselves are highly underdamped RLC filters, which induce substantial ISI. One communicates with system via WAV files, as described above, using command-line Python scripts. This system has been used successfully in a senior elective engineering course at Cornell for several semesters, for which course materials are available. The system can be used by anyone by using command-line python scripts.

Moving Forward

Extending the existing commcloud server to meet the overall objectives would require:

1. Two USRP B210 or equivalent radios. We currently have two B210 radios, although they are not yet connected to the commcloud. Having four radios would allow us to provide a wider range of different channels and also provide some needed redundancy. Cost: $1,119.00 per board plus $75 per board for an enclosure x 2 = $2388.
2. Antennas, cabling, and attenuators for the above radios. Cost: $500 (estimated).
3. Computers to drive the various channels. We have found that Mac minis work especially well for this. They are compact, and Mac OS X has proven to be a convenient operating system: it is POSIX compliant, which facilitates programming. the open-source libraries for the USRP work out-of-the-box on Mac OS X, and most professional-grade audio hardware does not have Linux drivers. Cost: $700 per computer x 4 = $2800.
4. Improved software infrastructure. The system will require an authentication mechanism, a web interface, basic performance monitoring, and a way for the different servers that comprise the overall system to coordinate. Cost: 40 hours of programming x $60 per hour = $2400 (estimated).
5. Professional system administration. Cost: $3000 per year (estimated).

This yields a total of approximately $8000 as a one-time charge and $3000 per year thereafter. Note that this does not include the cost of installing an exterior antenna or any underwater channels. Both MITRE and the ECE School at Cornell have provided funding for this project in the past. That funding has been almost entirely expended on building the current system, however. I am now contacting the IT Society, since I understand that the Society has some funding for “new initiatives," and I think this service could benefit the Society. Use of this service could even be a membership benefit, if we constructed it so that certain channels were only available to Society members. Beyond the IT Society, I am also contacting the US NSF, other US government funding agencies, and Mathworks.