

# Fishnet Assignment 4: An Audio Application

**Due: Monday Dec 9, 2002 at the beginning of class. Out: Friday Nov 22, 2002.**  
**CSE/EE461 Autumn 2002; Wetherall.**

The purpose of this assignment is for you to understand the structure of network applications by developing an audio application that communicates over the network by using the fishnet node you have already developed.

## 1 FishFone

Your job is to design and implement a FishFone application that provides a walkie-talkie service between IPAQs with the following features:

- *Amphibian Faux Sockets.* Your application should use the amphibian “faux socket” interface to communicate using your fishnet node. This is the same interface used by fishperf, the one sample application you have already seen. It requires that your application code be entirely separate from the fishnet node you have developed and run in a separate process than the fishnet node. Previously, you developed code on the fishnet side of the amphibian interface and included `<amphibian.h>`. This time you will develop code on the application side of the amphibian interface and include only `<amphibian_app.h>`. Further documentation on the real socket interface, on which amphibian faux sockets are modeled, is linked off the course web pages.
- *Transport Layer.* Your application should be able to both i) connect to a remote application to send audio samples over the network, and ii) accept connections from a remote application to receive audio samples over the network. To do this you should use the transport protocol you developed in the previous assignment. Your fishfone should accept connections on the FISH\_SERVICE\_FONE port as defined in `amphibian_app.h`. Each connection should only be used to send data in one direction, from the client initiating the call to the server accepting the call; you need to use multiple connections to have a two-way conversation.
- *Half-Duplex Audio.* Unfortunately, your IPAQ does not allow you to simultaneously play and record sound. To work around this, your application should support both operations, but only do one of them at a time. We suggest you detect whether a button is depressed on the IPAQ and use this as a user interface signal for switching between send and receive mode. That is, when a button is depressed, you should make establish a connection, take audio samples from the microphone on your IPAQ and continue sending them while the button remains depressed, and stop taking audio samples and close the connection when the button is released. Otherwise, when the button is not depressed, you should accept connections from other fishfones and receive samples from over the network, and send them to the speaker on your IPAQ while the connection lasts. We will supply you with code to help read and play audio samples and detect when buttons are pressed.

When you are finished, you should be able to call up your classmates on your fishfone and talk to them. To develop your application, you will probably want to tackle reading audio and playing audio separately. For example, try sending a known soundfile and get playback working before you handle recording audio samples. Note that, unlike previous assignments, you cannot run and test your code in a private fishnet because the development environment does not have the same microphone and speaker devices as the IPAQs.

## **2 Discussion Questions**

1. The fishfone as described above uses many transport connections. A simpler design would be to use only a single long-lived connection and send data across it in both directions. Describe the pros and cons of this design compared to the one above.
2. Describe which features of your transport protocol are a good fit to the fishfone application and which are not. Are the features that are not a good fit simply unnecessary, or are they problematic, and why? If problematic, how can we best deal with them?
3. Describe one way in which you would like to improve your design.

## **3 Turn In**

1. Turn-in all of your source code for your entire node electronically.
2. Bring a printed copy of your application code to class (you don't need to include the transport, routing and naming code from earlier assignments) along with answers to the discussion questions.
3. Provide us with a live demonstration of your fishfone running on your IPAQ with your fishnet node and communicating with our fishfone.

—END—