The purpose of this exercise is to experiment with audio compression and the effects of network congestion on the playout quality of an ADPCM compressed stream.

Write a half duplex ("one-way") audio phone that:

- reads 16-bit audio samples from an audio source (in Solaris, the output from /dev/audio is mu-law. You have to convert mu-law to 16-bit signed integer),
- performs the IMA ADPCM compression algorithm on this stream,
- transmit the stream over an UDP socket to a remote machine,
- decompresses and plays out the audio on the destination machine.

To write this program you will need to read data from and write data to, the sound card on Solaris machines. The general scheme for reading and writing is open the device /dev/audio, and perform standard file I/O on the descriptor for this file. The audio samples are read and written to the card as 8-bit signed mu-law integers. Additional details on using the sound card can be found in utility functions http://www.csc.ncsu.edu/faculty/rhee/clas/csc591E/program/audio. They contain audio open, close, read, and write functions. If you have any question on using the functions, send email to me or cheung@mathcs.emory.edu. You also need to link with libaudio.a and include files at /ncsu/dsresearch/CCF/SOUND/libmultimedia (you need to add dsresearch). For information about conversion between ulaw and linear sound can be found in ulaw2linear.h under /ncsu/dsresearch/CCF/SOUND/multimedia.

To receive full credit for the assignment, you must:

- implement a robust CODEC, in particular your system must perform "well" in the face of network packet loss. (Other network pathologies such as delay-jitter will be subject of future assignments and thus may be ignored for the moment.) Part of this assignment is for you to define the term "well".
- instrument your system to record performance statistics such as the number of samples generated, the number of samples received, average bit-rate (averaged over some suitable interval such as a second), average loss rate, etc.
- generate a brief, informal write-up documenting the operation of your CODEC. This write-up need only cover the implementation of your CODEC (e.g., you need not explain how ADPCM works). Think of this document as containing all the information for someone who knows the ADPCM algorithm would need to know in order to write a decoder for your coder.

For this (and all other programming) assignments you will turn in your program electronically using submit for grading.

The program should be neatly formatted (i.e., easy to read) and well documented. In general 75% of your grade for a program will be for correctness and completeness, 25% for "programming style" and documentation.