CSC 591E – Multimedia Networking
Homework 4 – Due April 12 Wed
TCP-friendly flow control

The goal of this work is to let you experience the behavior of end-to-end flow control for multimedia transmission over the Internet. In class, we discussed two TCP-like flow control techniques for multimedia streaming: TFRC and TEAR. Your task is to implement TEAR and TFRC and incorporate them into your ADPCM audio tool. Therefore, you need to build two versions of the audio tool; one with TEAR and another with TFRC.

In your implementation, TEAR (or TFRC) will give you the appropriate TCP-friendly sending rate. Your sending routine must adjust its transmission rate to at most the rate reported by TEAR (in almost all uncongested cases, your transmission rate would be less than the TEAR rate). For instance, if TEAR gives rate $x$ and your tool can send at rate $y$ at the maximum, the tool’s transmission rate must be $\min(x, y)$.

You can find a prototype implementation of TEAR (and TFRC) from the following sites (TFRC: http://www.aciri.org/tfrc/code/; TEAR: /ncsu/dsresearch/klarong/tear-source-code). The prototype is designed to transmit at the rate reported by TEAR (or TFRC). Your tool must send at a rate $\min(x, y)$.

Your task is to verify that your tools show fairness to TCP and their rate fluctuations are very low under most conditions. This can be done by performing the following experiments for each version of your tool:

1. Your tool must also be able to read from audio samples from a file. Your tool (this file version) must run under a Linux or Solaris OS. If you have an implementation on Windows, please find a partner who has an implementation on Solaris or Linux.

2. Run five instances of your TEAR tool with one second interval; each of the five instances runs in the following way.

    You have to run the sender part of your tool at a remote location (we will provide a Linux account in Korea. Please send email to Volkan to obtain one). The sender continually transmits encoded audio samples from Korea to a local machine in NCSU or your home machine from the beginning of the file and to the end of the file. After it reaches the end of the file, it loops back and starts from the beginning again. Repeat this process for 3 mins. The sender must record the the following data at every 100ms: the transmission rate, and the rate reported by TEAR (or TFRC). The receiver (which runs at a local machine) records each received audio sample and its reception time stamp in a file. The receiver must also record the packet loss rate and RTT. RTT can be estimated by periodically transmitting a probe packet (e.g., at every 100ms) to the sender and receiving an ack from the sender.

3. While you are running these five instances of TEAR, run other five instances of TCP connections; each sends a garbage data continually for 3 min. For each flow, record the sending rate.

4. Please make sure that the starting time of all the ten flow instances (5 TEARs and 5 TCPs) differ by one second (don’t start all of them at the same time).

5. Plot the sending rates, loss rate, and RTT of one TEAR connection and the sending rate of one TCP connection from the recorded traces over time (each point in the plot corresponds to one data sample...
at every 100ms). You can superimpose the sending rates of TEAR and TCP in the same graph. Thus, you need to have 3 graphs.

6. Replay the received audio samples of one TEAR connection according to their receiving timestamps. Listen to the sound, and report the relative sound quality score in a scale of 1 to 10.

7. Repeat the steps 2, 3, 4, 5, and 6 with the TFRC implementation.

8. Repeat the above experiments at every hour for two days. You need submit 48 graph sets and trace sets (each set consists of 3 graphs) for TEAR, and 48 sets for TFRC.

Submit the assignment electronically before the due date using submit utility.

Remember the program should be neatly formatted (i.e., easy to read) and well documented.