Systems Programming: Assignment 4 Program Description

Kevin James Waller

October 21, 2018

1 Protocol Description

First, the client sends the filename to the server. This packet is assumed to arrive at the server.

When the server receives the filename, it reads the audio header information from the requested audio file, such as sample rate, sample size and channels. It sends this in a specially formatted message to the client. The server then starts waiting for an acknowledgment from the client.

The client receives the audio header information, and sends a simple "ACK" message to the server.

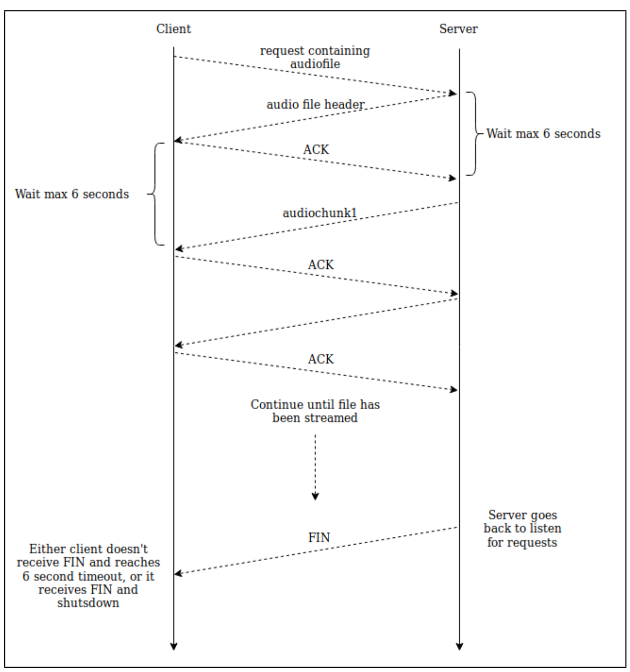
Once the server receives this ACK, it starts reading chunks of the audio file, and sending these over to the client. After sending each audio chunk, the server waits for an acknowledgment before proceeding to read and send the next audio chunk.

The client simply receives an audio chunk, and sends an ACK back. Whenever an audio packet gets lost on its way to the client, the server will notice that it has not received an ACK back. It will then continue to keep sending the same packet for a maximum of 6 seconds, until it closes the connection and goes back to listen for requests.

Whenever an ACK gets lost, the same thing happens. The server wont get an ACK, and will resend the packets for a maximum of 6 seconds.

Once the entire audio file has been streamed, the server sends a FIN message to the client, and starts listening for other requests.

When the client receives the FIN message, it shuts down. In case the FIN message does not arrive, the client still has a timeout of 6 seconds, at which it will automatically shut itself down.



2 Maintaining Bitrate

To maintain the proper bitrate, the server must send packets at the right transmission rate. The goal is to find out how long we need to let the audioserver sleep in between each audio chunk transmission.

$$Bitrate = samplerate * samplesize * channels$$

Since the server is sending audio chunks in bytes, it is more convenient to get the byterate.

$$Byterate = samplerate * (samplesize/8) * channels$$

The byterate is the amount of bytes that must be transmitted per second in order to maintain proper throughput for the audio file to play at normal speed. Thus, to calculate the amount of packets we need to send per second, we simply divide the byterate by our packet size, or the size of the audio buffers we send over each packet in bytes.

$$packetspersec = (byterate/BUFSIZE)$$

However, packet per second still needs to be converted to how long we need to sleep.

$$Timeout = 1/(packetspersec)$$

This timeout can be converted to either microseconds for the "select" function and nanoseconds for the "nanosleep" function in order to maintain the bitrate.