Real Time Voice Communication assignment

Abstract:

The main goal of this assignment is to establish a voice call between 2 different systems(laptops) over internet. In order to have good user experience, this call should be of real time(less end to end delay). So, we are using UDP(User Data Protocol) of transport layer because its doesnt do re-transmisions unlike TCP. These re-transmissions incur more delay. Therefor we use UDP instead of TCP. But the drawback in UDP is, its not reliable unlike TCP. There is sequence numbering in TCP, thereby it can acheive reliability and can even sort packets which were received out of order. There is no such sequence numbering in UDP. Here we are using socket programming for sending and receiving packets. We are making use of pulse audio library for recording and playback.

Implementation:

We created a socket through we send and receive data packets. We created two threads. One thread is send thread. In this send thread we record the audio of the user using pulse audio APIs in to a buffer and send this buffer containing audio packets through the socket we created to other user(laptop). Second thread is receive thread whcih receive packets from other user and play the audio data packets received using pulse audio APIs. These two threads run forever independent to each other. We created two different handles, one for playing the received packets and one for recording the audio to a buffer. Here we are sending the un compressed data which requires more bandwidth.

Challenges faced:

1. **Unwanted glitches while playing the audio:** In the beginning, we heard so many glitches while playing the audio packets received from other user. These glitches are because of creating a new handle(source for playing) and freeing the source every time we receive a packet. We overcome these glitches by making use of only one source(handle) for playing all the data packets.We are not freeing this source(handle).

2. **Choosing sampling rate:** In the pulse audio example, they used the rate(sampling rate) as 44100 because the maximum frequency component of music is around 20000. In order to reconstrcut(to hear the signal back properly), the sampling frequency should be atleast twice of maximum frequency component(Niquist criteria). But here we are making voice calls only. The maximum frequency of human voice signal is 3400. Therfore we are using sampling rate as 6800(twice of max freq component) thereby reducing high frequency noise components effectin our voice signal.

Improvements that can be done:

1. By using suitable data compression techniques we can compress data before sending to other user, thereby bandwidth is effectively used.

2. By using proper noise cancellation(reduction) techniques we can reduce the noise, thereby give a better user experience

3. By not sending the silent packets(i.e no user voice data, only noise is there). Thereby bandwidth is effectively used and can provide good user experience.

4. Making it a multi-conference call from a 2-way call.

5. Provide a good Graphical User Interface to the users.

Limitations:

1. Both the users(laptops) should be under the same network.