Adnan Salihovic and Jacob Boicken

Pre-Lab

Write the code to create a UDP socket and bind it to the IP address of the machine your program is running on, and to a given port number.

- This code will be submitted in the zip file with our other program.
- Shows binding to our 127.0.0.1 address

Exercise

1.) What was the effect of increasing the loss rate on the video quality?

The loss rate made the video almost never update, even at a 5% loss rate. It showed about the first few frames, then it froze and held for a few seconds before skipping to a much later frame in the video. Our thought is that the information sent isn’t an entire frame but what to update for a frame of video. So when the information mismatches in order, vlc waits until it gets a new frame to update from or until it gets the packets to use to update.

2.) To counter the effect of network loss, someone suggested using TCP instead of UDP for multimedia communication. Do you agree? Why? Why not? (Answer the question in terms of buffer space, jitter, and retransmission time.)

We would agree that this ensures the entire video is received and rendered. However, since TCP is connection oriented, has error handling in the protocol, and has a slightly larger header size. This would mean the overall delay is much larger for each frame / update being sent, since a back and forth connection needs to be established, validation must occur of the packets, and each packet will take a larger buffer and more time to read. This delay may cause the video to jitter and freeze, since, if the transmission is slower than the rendering of the video, then the playback needs to wait for the packets before rendering. This would be more like the case with TCP. Additionally, since TCP wants to ensure every packet is received, the video may buffer as it waits for a loss packet to be retransmitted. This creates a large time when the video would be frozen, which I believe would be after multiple ACKs being sent back to the sender. With it as UDP, the video will just skip things that are lost so that it is always playing.

What We Learned

In this lab, we learned how to program a UDP socket. We also learned how to create a connection between a source and a destination by using sockets, bind() from the sockets API. We also learned how to use the recvfrom() function to capture the VLC video on the source computer then, using the sendto() function, we were able to send the VLC video to the destination computer so that the destination's VLC application played the same video. Lastly, we learned how to set up loss rate so that when the argument was inputted by the user, the packets traveling from source to destination would be randomly lost based on the percentage inputted.

Code submitted in .zip file, and lab was demoed during lab hours.