

Lab 3 Report

By: Joseph Schmidt

Exercises

- 1) Increasing the loss rate meant that there we were increasing the amount of packets we would not send. For our case we did a random number modded it by 1000 and if it was greater than or equal to the loss rate it would send. So the higher you increase the loss rate the higher lower the bounds on the available numbers for the random number to make it send. So as the loss rate increased it decreased the chance of the packet sending.
- 2) No, you should stream in UDP. UDP is a connectionless protocol and continues to send data while TCP guarantees quality and no packet loss. This means TCP will have a will resend every time there is a packet loss. Everytime it resends it takes time which is called retransmission time which affects the quality of the stream. TCP will have a more consistent jitter though as TCP has flow control to make a more consistent delay. In terms of buffer space TCP requires more buffer space as it provides data in an ordered and reliable way. This means it will hold onto packets that arrive out of order increasing the buffer size. Overall it is better to use UDP because it reduces the buffer size and doesn't require retransmission time.

Conclusion

In this lab I learned alot about streaming, UDP and socket programming.

In this lab we were able to set up a UDP relay server. It was interesting to be able to pass along packets to better understand how UDP works. In order to create this UDP server we had to make two sockets on two different ports. Next, we had to specify that the socket was using IP and the IPs we wanted the socket to interact with. We then bound one of them for receiving from a source destination. Next I just created an infinite loop. In there, I started off by having the `recvfrom` function that took in the socket I created for the source and a buffer to write the data to. I then had an if statement which was used to implement our artificial loss rate. It worked by making it the higher the loss rate the less numbers you could choose from `rand()` to go into the if statements. It basically worked by decreasing the chances that you get into the if statement the higher the loss rate. Inside of the if statement there was a `sendto` method that sent a buffer of data to the other socket we had created.

With this experiment we could see at about what rate did it start to affect video quality and what happened when packet loss occurred. From our experimentation this occurred at around .5% - 5%. When I increased the lossy rate you would see weird pixelation errors where some of the screen would be still and other parts would be moving, lag, jitteriness, and discoloration. At 10% I was only able to see one frame and it was stretched from the bottom half down.

Overall I learned about UDP and how it works by sending packets at a target IP without a handshake and without the knowledge if that ip is listening. It also doesn't work to resend back packets as it sends packets and it doesn't hear back like in TCP. This lab was great for looking at the difference between TCP and UDP connections.