watterso - Lab6p3

The basics of the negative ack are obvious. The sequence number should be incremented (reset to 0 when it hits the max value) and included in the sequence number portion of the packet. And the client responds with the negative ack as expected.

This particular implementation however is trying to juggle congestion control and guaranteeing packet delivery all while maintaining a pseudo-real time stream. The best way to solve this problem is to not guarantee delivery, except when packet-loss build up occurs, packet loss in streaming is acceptable to users. However given that the problem is explicitly asking for an ack system, I will continue.

The implementation becomes a little more complicated when the negative ack is received on the server side and mixed in with the congestion-control / streaming logic. The pseudo-realtime nature of this problem simplifies the design process. Because packets are time sensitive, when the server receives a negative ACK, it will immediately send the packets requested. If this proves to still be susceptible to congestion problems another improvement can be made on the client side. In the feedback packet the values of Q(t) and Q* can be adjusted to account for the lost packets and their retransmission. The key is to maintain the dumb sender/smart receiver model in the base problem and not added complicated logic to the server.