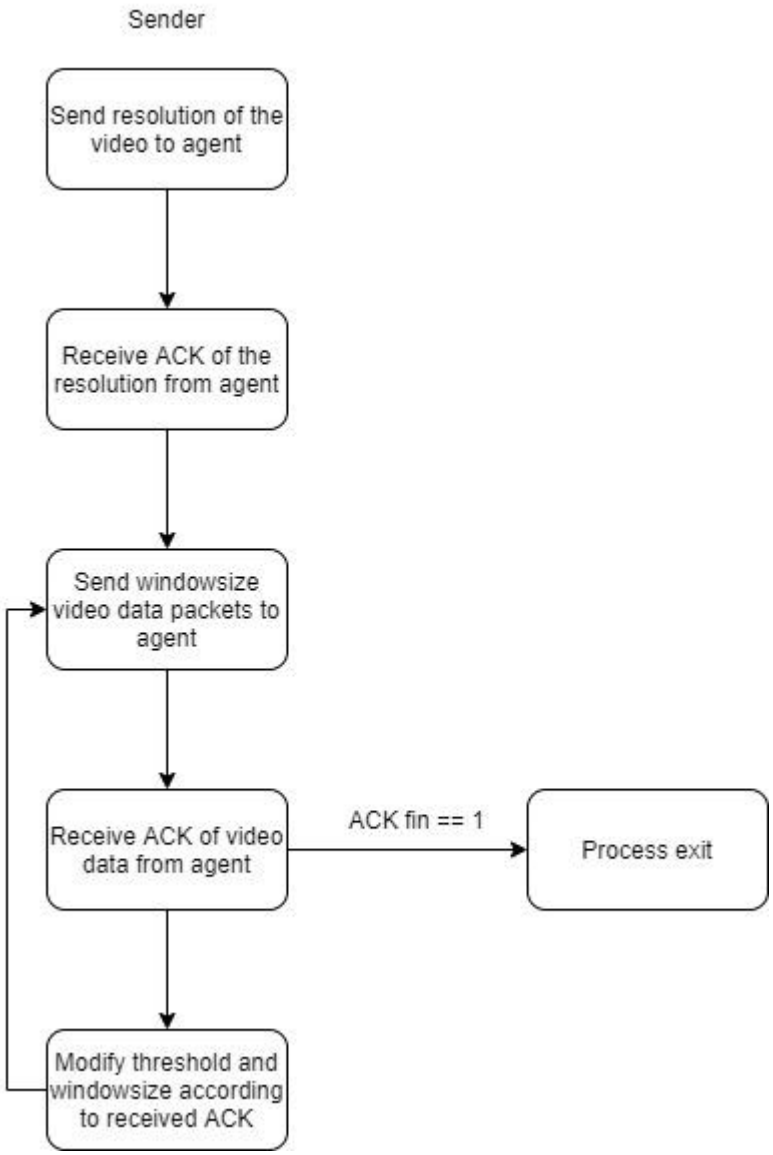
How to execute :

Input of agent : ./agent <sender IP> <receiver IP> <sender port> <agent port> <receiver port> <loss_rate>
Input of receiver : ./receiver <receiver_ip> <agent_ip> <receiver_port> <agent_port>
Input of sender : ./sender <sender_ip> <agent_ip> <sender_port> <agent_port> <filepath>
For example :
$ make
$ ./agent 127.0.0.1 127.0.0.1 8887 8888 8889 0.3
$ ./receiver 127.0.0.1 127.0.0.1 8889 8888
$ ./sender 127.0.0.1 127.0.0.1 8887 8888 ./filepath

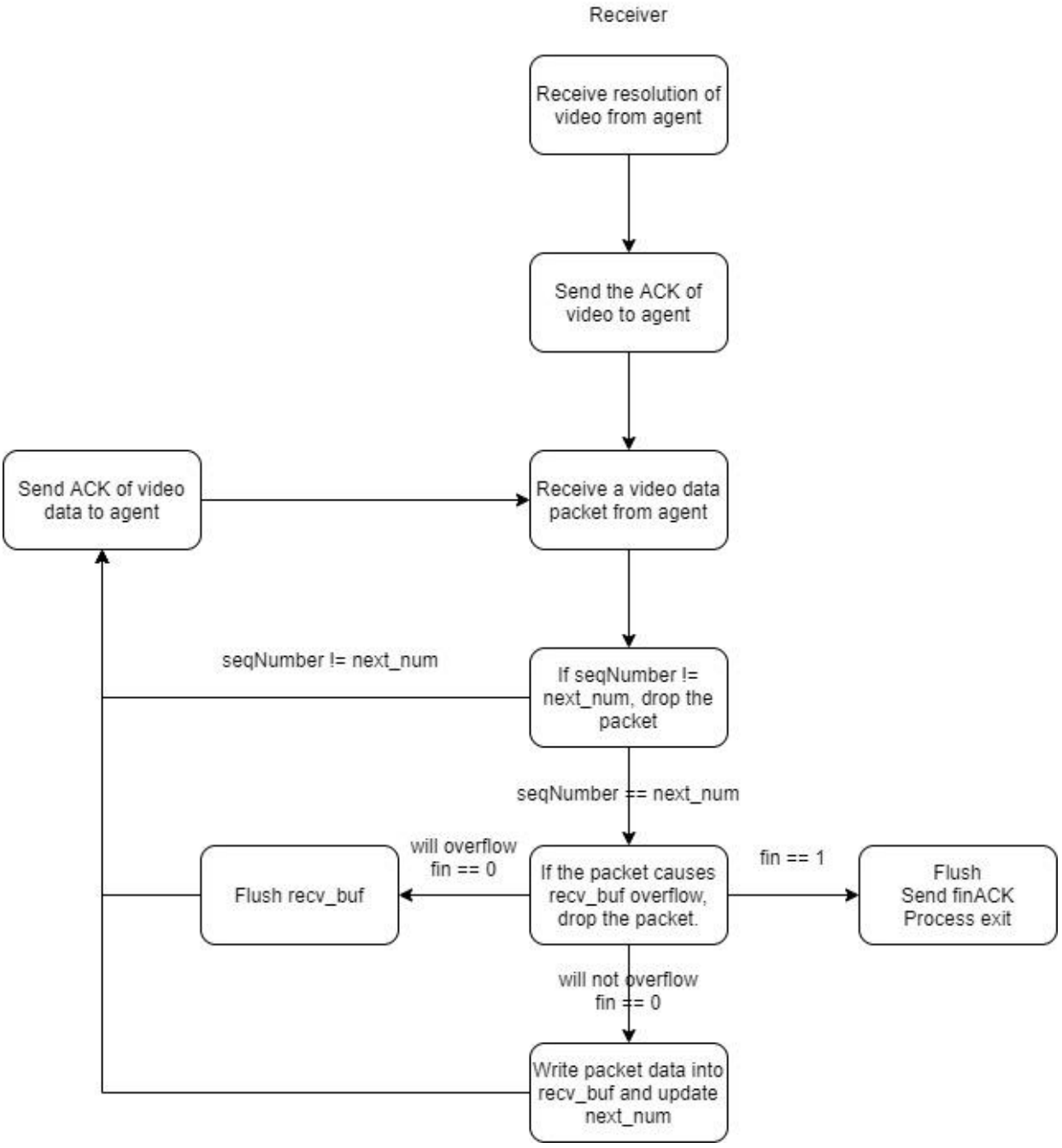
Program structure :

Sender :



Sender will send resolution of the video to agent, and then wait the ACK. After that, sender starts to send video data.
Sender will send windowsize packets to agent. After sending a packet, sender will non-blocking receive ACK and move the sending window by adjusting ack_num according the received ackNumber of ACK packet.
After sending windowsize packets, sender will receive ACK in 0.5s.
If the received ackNumber == next_ack, all packets are received and thus increase windowsize.
If the next_ack ACK packet is not received, some packets are lost and thus decrease threshold and set windowsize to 1.
After receiving ACK and modifying windowsize, sender starts to send packets from the last unACKed packet.
When there is no more frame data to send, an empty packet with fin = 1 will be sent. After sender receive finACK replied by receiver, sender will exit.
If sender does not send packet with fin = 1 but receive finACK, then sender will assume that receiver wants to stop and sender will exit.

Receiver :



Receiver will receive resolution from agent, and then send ACK to agent. After that, receiver starts to receive video data.

Receiver will receive one packet at a time, then receiver will check whether the seqNumber == next_num.

If seqNumber != next_num, receiver simply drop the packet.

If seqNumber == next_num, receiver will check fin of the packet and check whether recv_buf will overflow after write the packet data.

If fin == 1, then flush recv_buf, send finACK to agent and then exit.

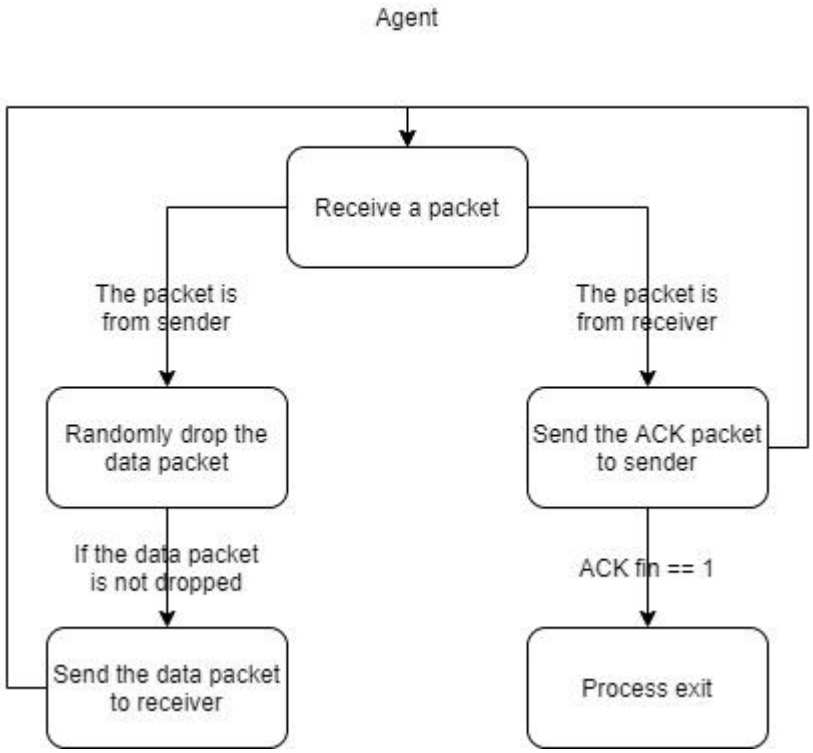
If fin == 0 and recv_buf will overflow, receiver will flush recv_buf and drop the packet.

If fin == 0 and recv_buf will not overflow, receiver will write data into recv_buf and update next_num.

After process the packet, receiver send ACK to sender.

When press ESC, receiver will send finACK to agent to tell agent and sender to stop, then receiver exit.

Agent :



Agent will receive a packet and determine where it comes from.

If the packet is from sender, then agent will randomly drop the packet according to given loss rate.

If the packet is not dropped, then it will be sent to receiver.

If the packet is from receiver, then agent will send the packet to sender.

If the packet's fin == 1, then agent will exit after send the finACK to sender.