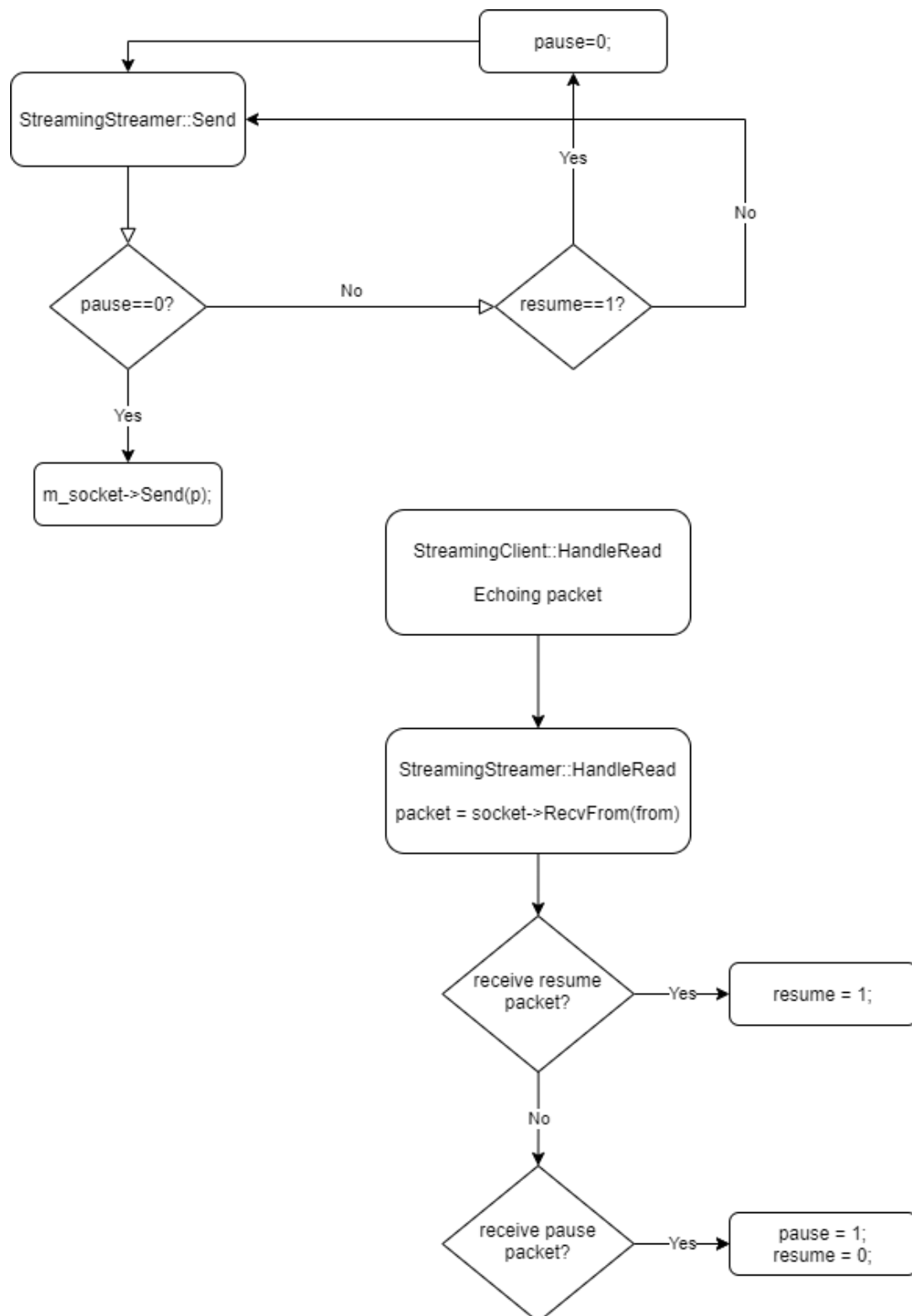
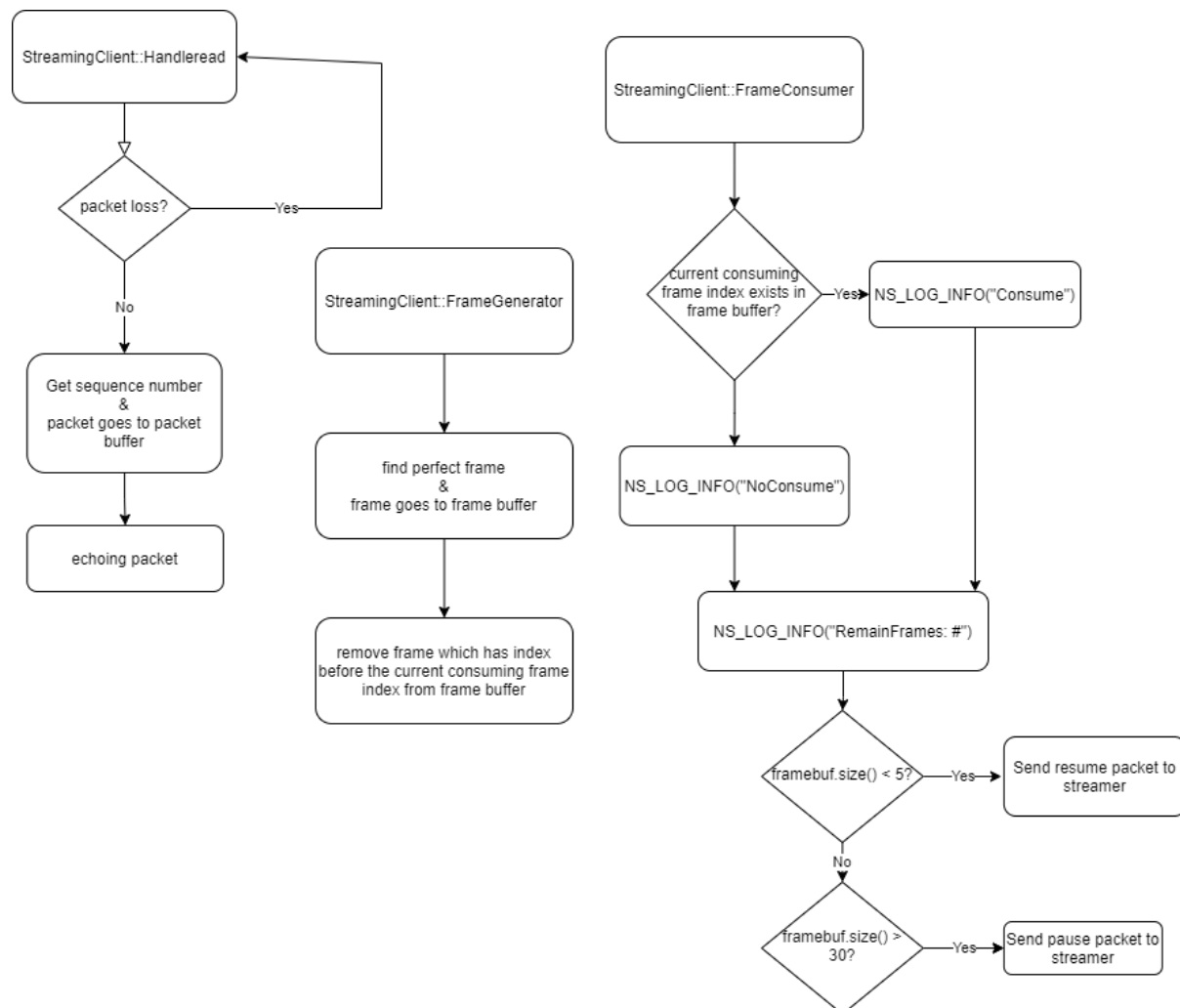


[Question1, 25pt] Explain the structure of the streamer and client implemented by using a flowchart.



In Streamer structure, there are two big functions named Send, and HandleRead. In Send function, if pause packet is not arrived from client(pause==0), just send frame consisted of 100 packets to

client. If not(pause==1), check the value of resume. If we can send packets because resume packet is arrived from client(resume==1), make pause = 0 to send packets at next call. If not, just do nothing. Next, in Handleread function, receive echoing packets from client. If we receive resume packet, make resume = 1 to resume sending packets. If not, check if we receive pause packet. If so, make pause = 1, and resume = 0. Because we should pause sending.

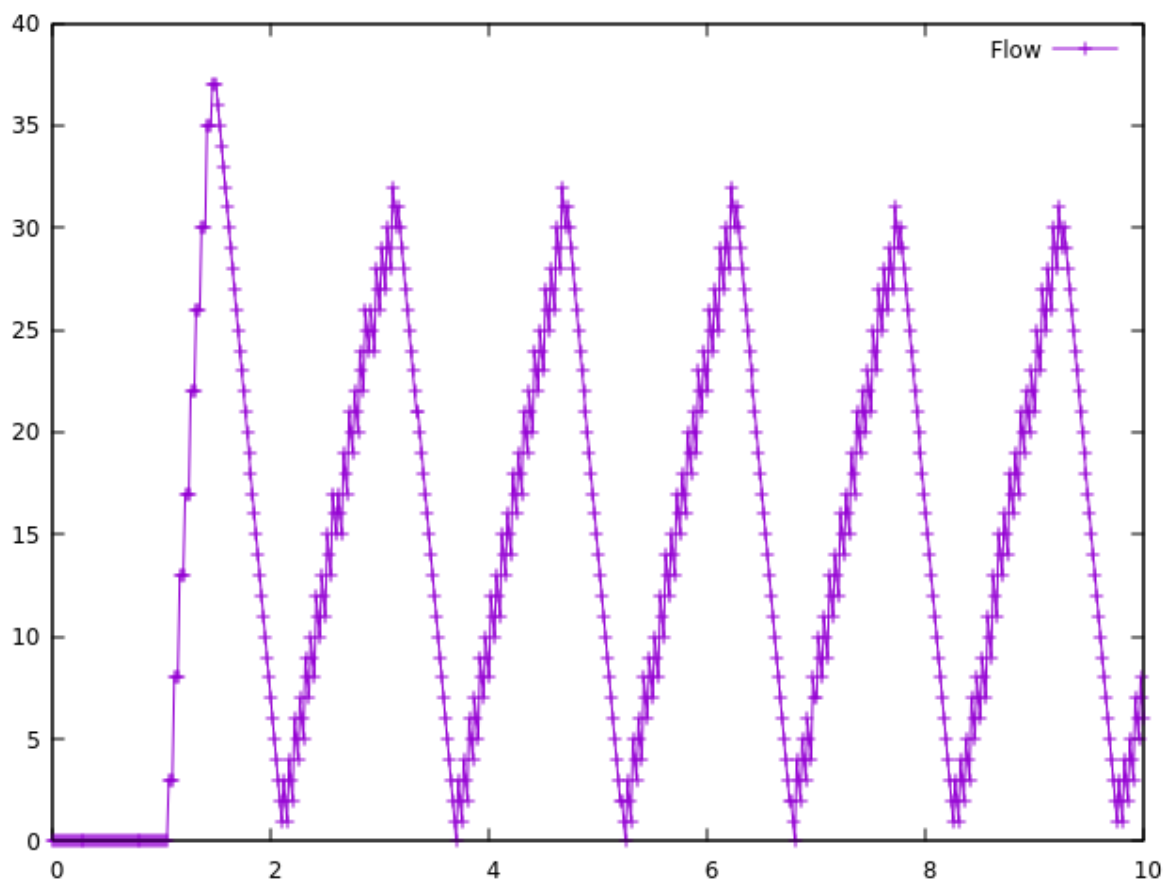


In Client structure, there are three big functions named Handleread, FrameGenerator, and FrameConsumer. In Handleread function, decide whether received packet is processed to be a loss packet by predefined loss rate. If processed to be a loss packet, just skip all the remaining actions like removing header, echoing packet. If not, get sequence number from packet's header, and packet goes to packet buffer with proper frame index. Then, echo packet to streamer. Next, in FrameGenerator function, find perfect frame which means 100 packets which consists a frame is all gathered in packet buffer. Perfect frames go to frame buffer. Then, remove useless frame from frame buffer. For example, when at specific timing like 1.5seconds, FrameConsumer consumed Frame #120,

if then, frames which has the frame index before 120 such as #100, #90.. should be removed from buffer. Lastly, in FrameConsumer function, try to consume frame sequentially. If the frame which currently should be consumed exists in frame buffer, log the info("Consume") and consume the frame. If not, also log the info("noConsume"). Then, log the info about the number of remaining frames in frame buffer. After logging, seek the size of frame buffer. If size is smaller than 5, send resume packet to streamer to resume sending frames. If size is bigger than 30, we can expect that buffer goes full, so send pause packet to streamer to pause sending frames.

[Question2, 25pt] Graph the number of frames remaining in the buffer every 1/60, and explain the FrameBuffer condition (Using Figures).

**x-axis of following graphs : time(s), y-axis : the number of remaining frames**

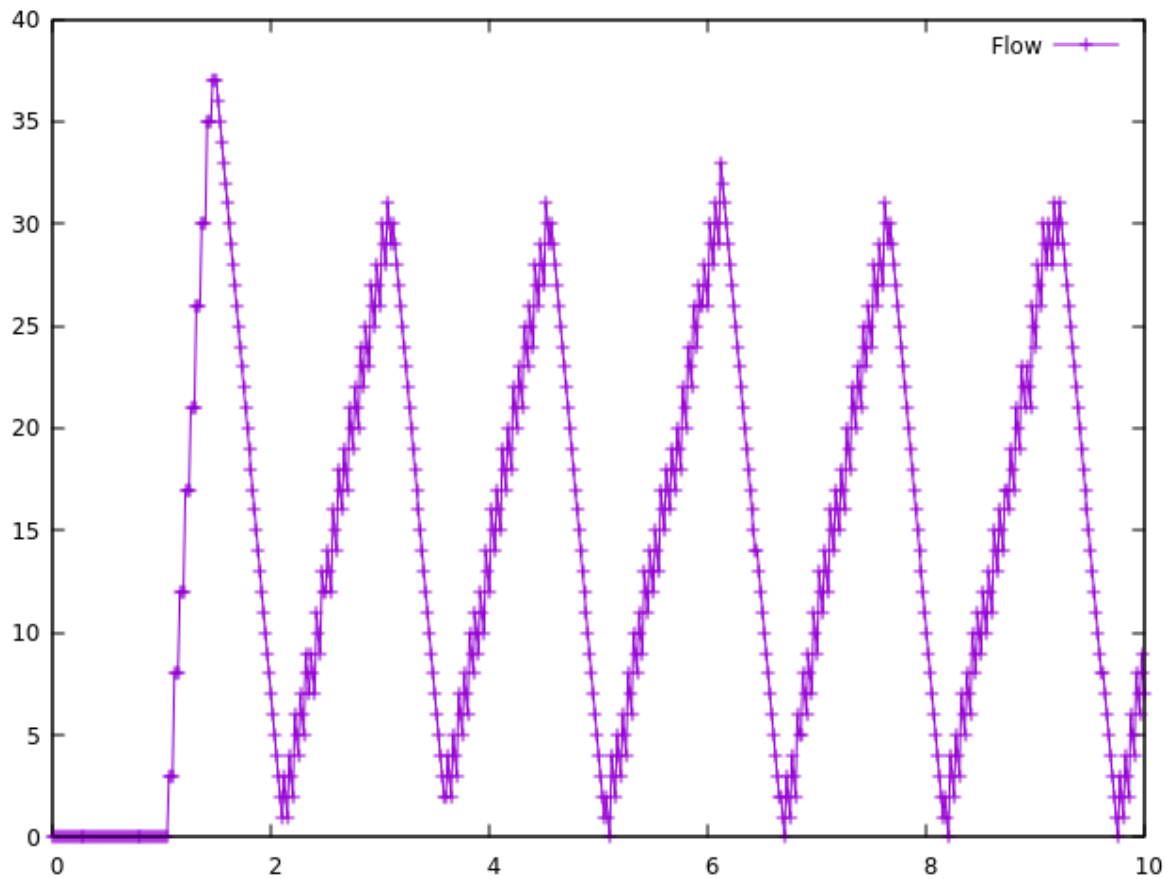


FrameBuffer firstly increases until about size = 37 and start to decrease until size = 1. Then repeatedly increasing until about 32, and decreasing until 1 or 0. Because resume packet and pause packet take some time to arrive at Streamer::Handlerread().

[Question3, 25pt] Explain the FrameBuffer condition while adjusting the RxNoiseFigure between 45.0 and 45.5 (Using Figures).

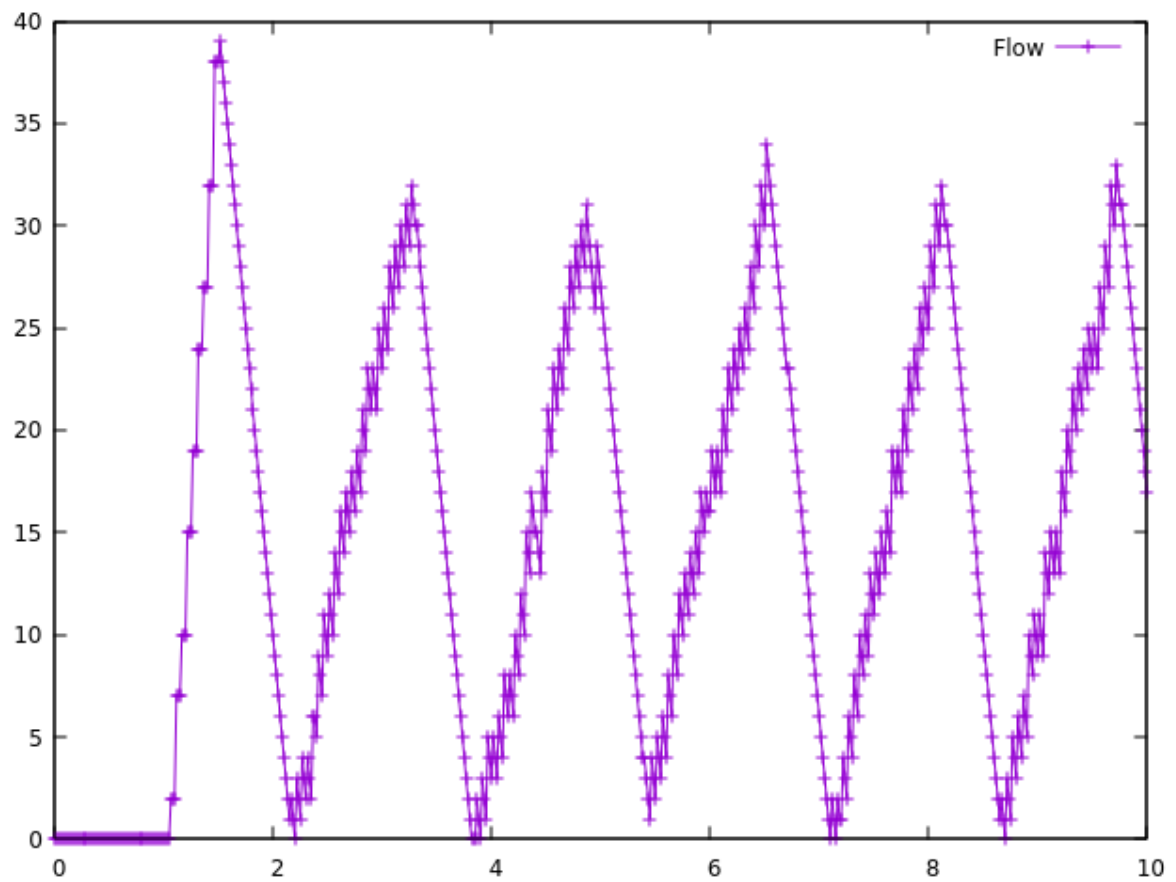
If RxNoiseFigure = 45.0, graph is same as graph in Q2.

➤ RxNoiseFigure = 45.1



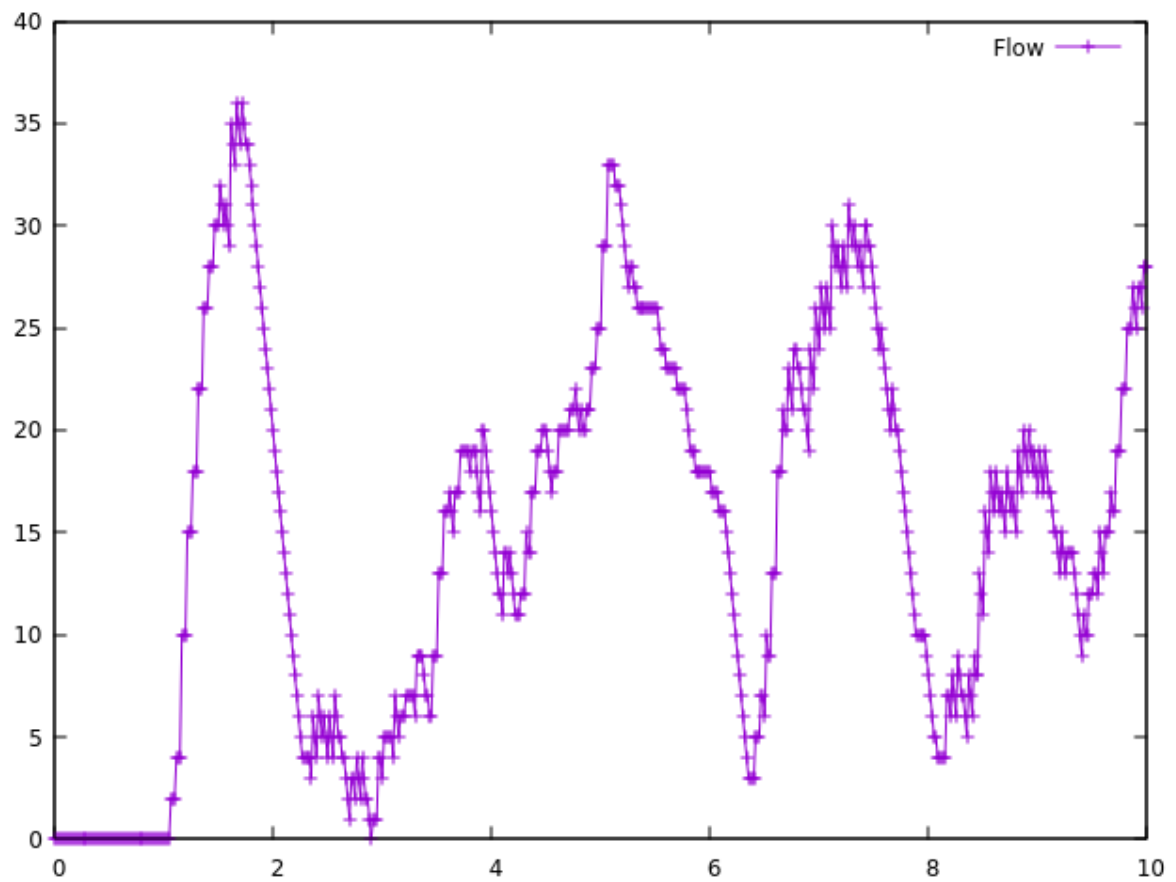
The graph is similar with the graph that RxNoiseFigure is 45.0. But at the timing near 3.8s, Frame buffer size decreases until 2, while decreases until 0 in the graph RxNoiseFigure is 45.0.

➤ RxNoiseFigure = 45.2



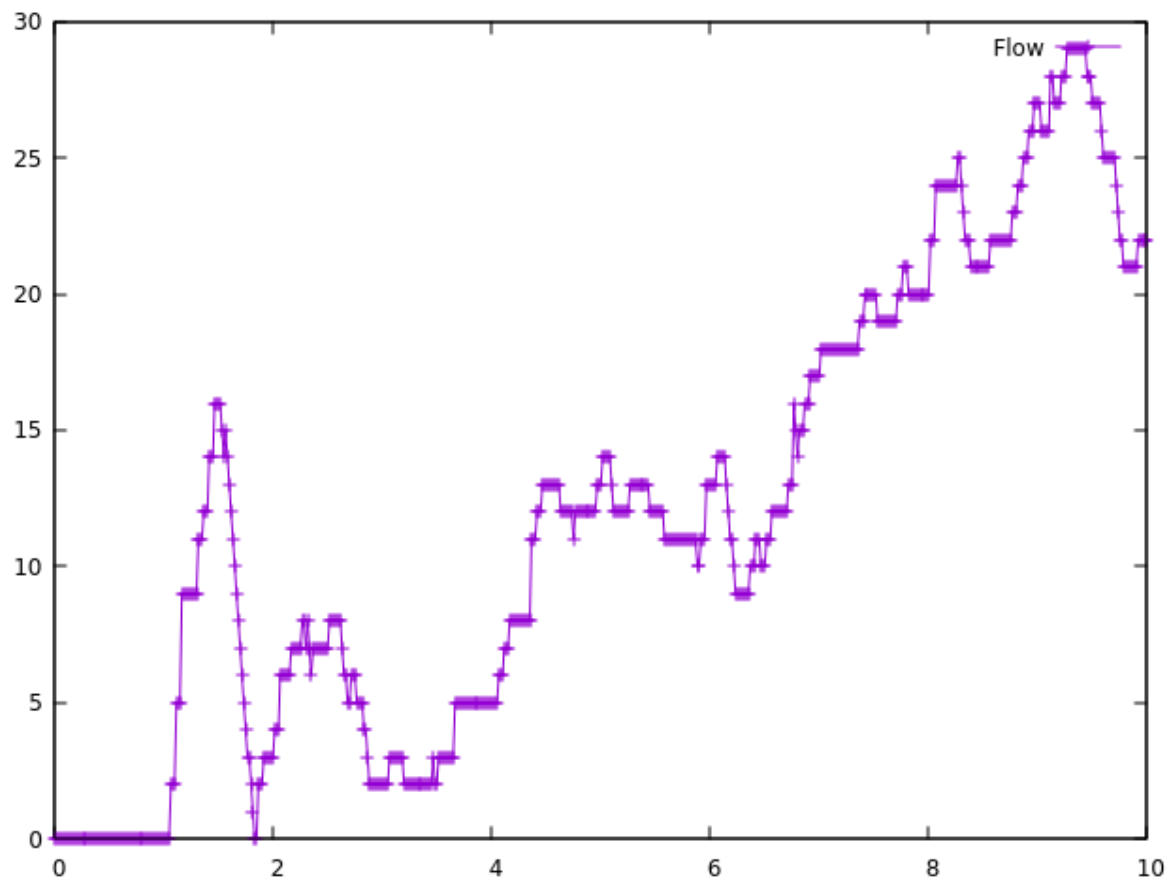
The shape of graph is similar with the graphs before. But the speed is a little bit slow down. Compared with 6<sup>th</sup> top point of graph 45.1 vs 45.2, you can see that 6<sup>th</sup> top point is taken about 0.7s later.

➤ RxNoiseFigure = 45.3



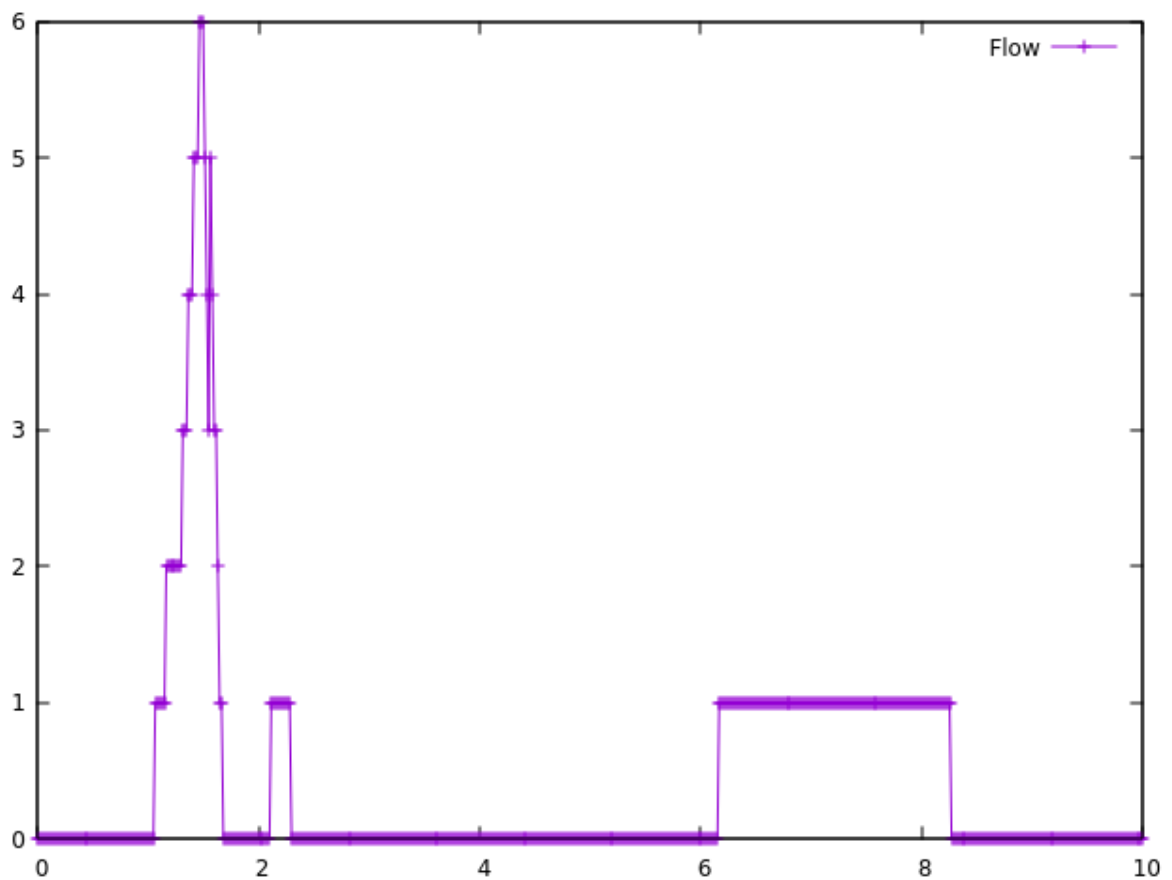
After 45.3, graph goes weird. It takes much longer to hold more than 30 frames in buffer compared to graphs before. The graph shows that there are some troubles to send or make frames.

➤ RxNoiseFigure = 45.4



The troubles are getting worse, so in this condition, it takes almost 10 seconds to go to the top point that means more than 30. What's special is, at timing near 1.8s, the number of frames in buffer decreases until zero although pause packet is not arrived.

➤ RxNoiseFigure = 45.5



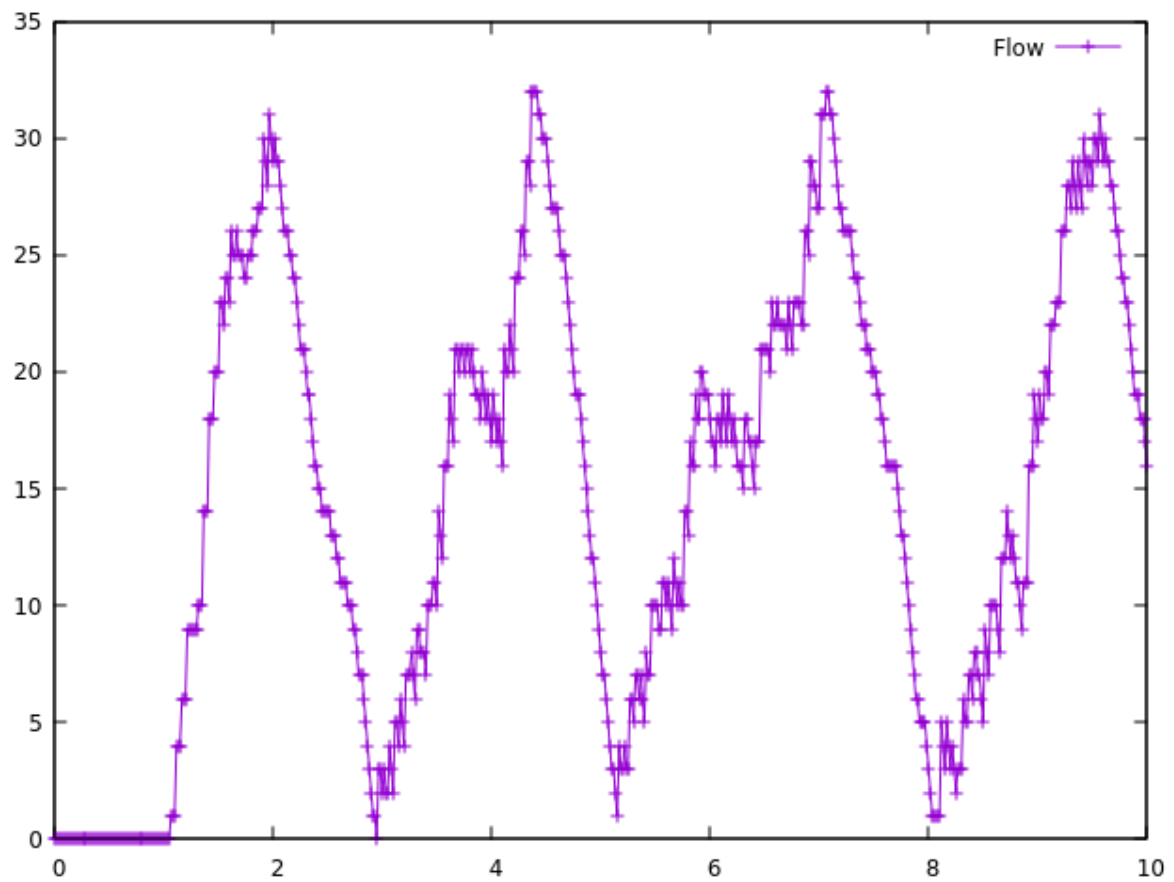
In this condition, FrameGenerator seldom makes frames. I think it's almost impossible to consume proper index of frame.

[Question4, 25pt] Modify the code to force packet loss in the Client's HandleRead part, and explain what happens in FrameBuffer according to the packet loss rate (Using Figures).

I fixed value of RxNoiseFigure = 45.0.

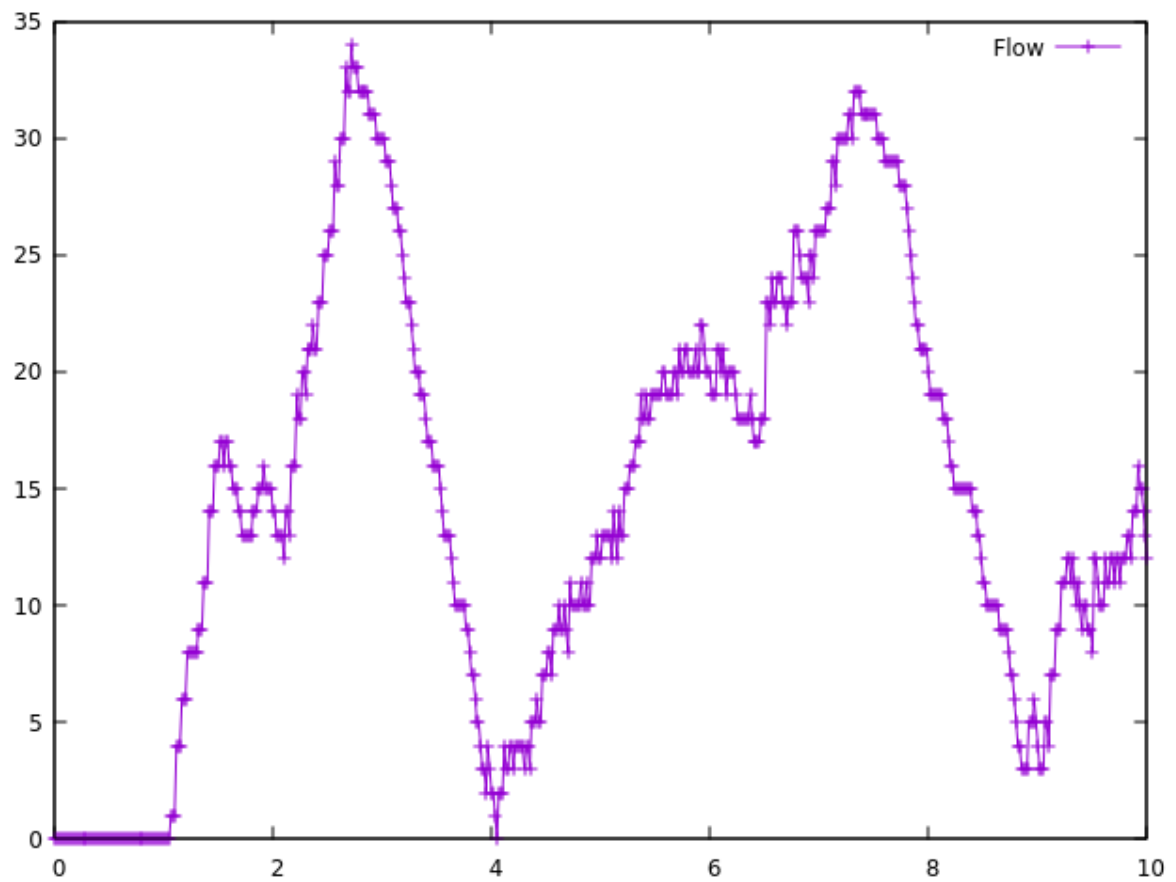


➤ Loss rate = 0.5%



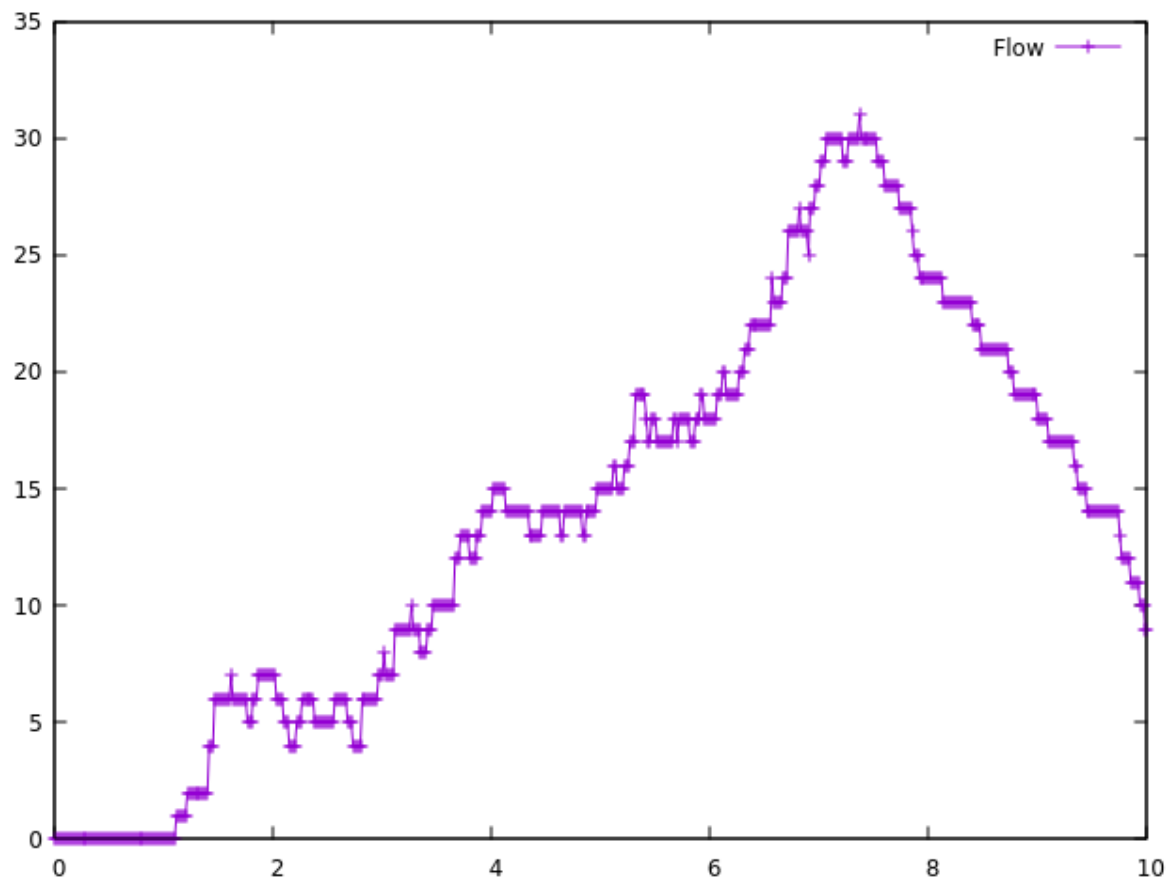
Sometimes it can't make frame because of packet loss. It's frequency is very low, so frame consuming is quite stable. If packet loss is happened, like the shape at timing near 4s, the number of remaining frames just decreases for a while because it just consumes frame not makes.

➤ Loss rate = 1%



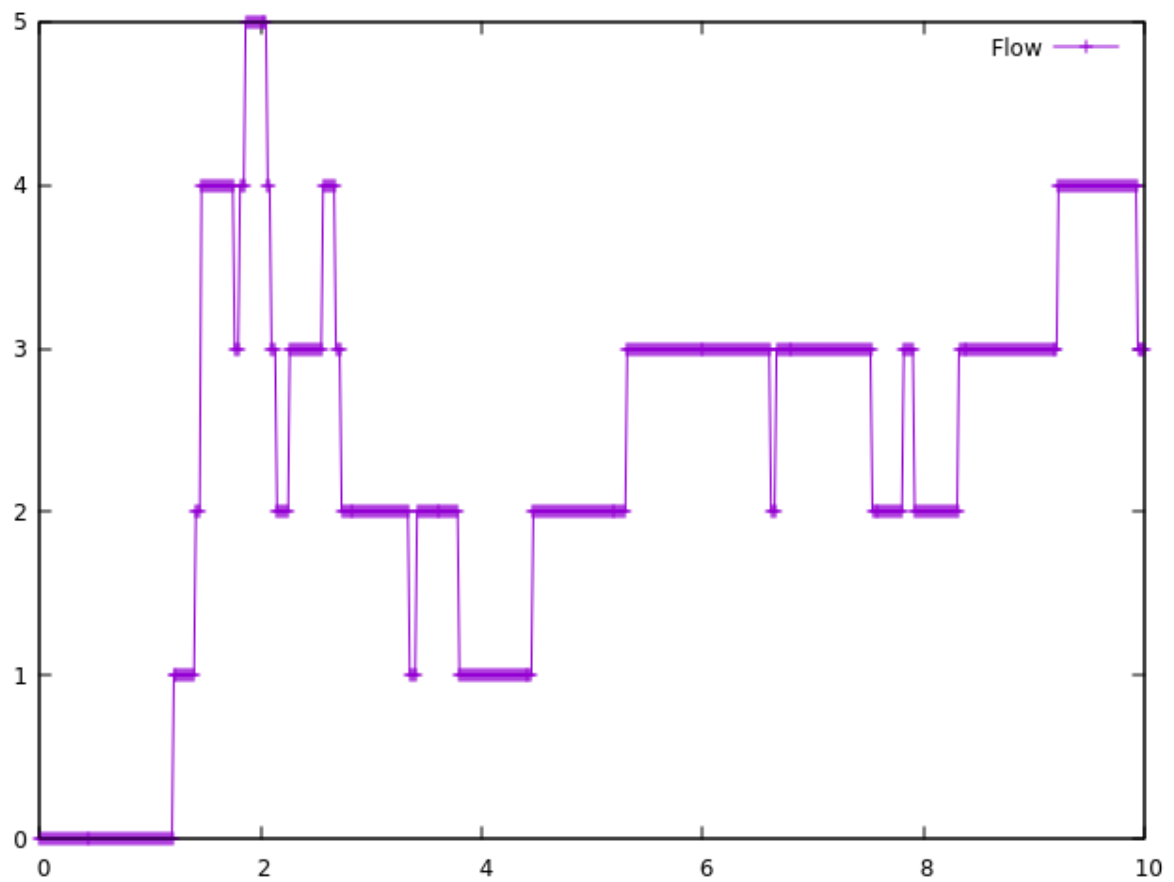
Packet loss frequency goes a bit higher, but still the number of remaining frames in buffer goes up to more than 30 although very slow.

➤ Loss rate = 2%



Frames are stacked in frame buffer more slowly. Frame buffer size goes up to 30 once at near 8 seconds. What's special is, when we see the part of timing 8~10s, the speed of consuming frames also go slower.

➤ Loss rate = 4%



Frame making speed and consuming speed is almost similar. The number of remaining frames remains near 2~4.