**ABP**

The code is similar to the flow chart given by the lab instructor. There are ES\_processor, which processes the returned packets. The send function is called to send the packets to the receiver side. Send function is composed of 3 parts, forward channel, receiver, and reverse channel. Channels calculate whether the packet is lost, corrupted, or safe. Then the send function summarizes all the information in a struct and put the struct in \_ES, which latter is recovered by ES\_Processor. Then the ES\_Processor, depending on the type of the event that came back, either process it or time the system out.

**ABP\_Nak**

The code is mostly same as ABP. The only difference is that when the event comes back in \_ES, instead of ignoring the erroneous packets, it is treated with the similar process as the time out event. The time is synced with the packet time. (Since the event is error event, the time is not increased by delta.) The other procedures are pretty much same.

**GBN**

It is a bit different from each other process. ES\_Process has been removed, but the Send() function is now wrapped with Sender function, which now processes the packets that are returned. The functionality of Send() function is same. In sender function, first it puts time out event into \_ES. Then the Send() function is called. When it returns, the Sender function analyzes the response. If it is time out, the buffer counter becomes 0. If it is ack without any error, the window slides in accordance to RN and new packets are added to buffer.