Implementation of ABP:

The design logic is from figure3.10 of the textbook.

A_output will be called from layer5, passed the data to be sent to other side and start the timer. Then the A_input will be called from layer3 when a packet arrives for layer4 to validate if the packet is not corrupted the timer will be stopped. A_timerinterrupt will be called when A's timer goes off in order to resend the packet when the timeout occurred. The B_input will be called from layer 3 when a packet arrives for layer 4 at B side. The send size A has two states. In the leftmost state, the send-side protocol is waiting for data to be passed down from the upper layer. When A_output occurs, the sender will create a packet containing the data to be sent, along with the checksum. In the rightmost state, the sender protocol is waiting for an ACK from the receiver B side. If an ACK packet is received, the sender finds packet received correctly and thus the protocol returns to the state of waiting for data from the upper layer by calling flipDigit. When the sender is in the wait state, it cannot get more data from upper layer. It must receive ACK and change the state.

Implementation of GBN:

The design logic is from figure 3.20 of the textbook.

Dfine base to be the sequence number of the oldest unacknowledged packet and nextseqnum to be the smallest unused sequence number. Sequence numbers in the interval [0,base-1] correspond to packets that have already been transmitted and acknowledged. The interval [base,nextseqnum-1] corresponds to packets that have been sent but not yet acknowledged. Sequence numbers in the interval [nextseqnum,base+windowsize-1] can be used for packets that can be sent immediately, should data arrive from the upper layer. Finally, sequence numbers greater than or equal to base+windowsize cannot be used until an unacknowledged packet currently in the pipeline has been acknowledged.

The sender starts with base=1, nextseqnum=1. A_output will be called from layer5, passed the data to be sent to other side and start the timer. Then the A_input will be called from layer3 when a packet arrives for layer4 to validate if the packet is not corrupted the timer will be stopped. A_timerinterrupt will be called when A's timer goes off in order to resend the packet when the timeout occurred. The B_input will be called from layer 3 when a packet arrives for layer 4 at B side. It validates if the message is good then deliver the data and send the ack back to A side. B side need maintain is the sequence number of the next in-order packet. This value is held in the variable expectedseqnum=1. The limitation are the window size and the buffer size.