Say you want to transfer 1Kbyte (i.e. 8000 bits) frames over a 0.5 Mbps channel with a one-way delay of 32 ms, using the sliding window algorithm, with SWS = RWS.

a) Compute the value of SWS that will ensure high channel utilization.

Assume that RTT = 2 * Tp

b) Suppose the sender will time-out at 1.5*RTT, and an ACK frame is sent for each correctly received, in-order frame (Ignore the transmission time of the ACK frame but do not ignore the propagation delay). **Sketch a timeline diagram for the following case**, labeling each data frame and ACK as F_1, F_2, … and ACK_1, ACK_2, … etc. (assume that there are enough bits in the sequence field in the header of the frame such that there is no need to wrap-around sequence numbers). Clearly label the time axis assuming the first frame F_1 is transmitted at t = 0. No ACKs are sent for out-of-order frames.

CASE: F_2 and F_4 are lost (remember, frame lost means it did not reach the receiver)

We want to continuously be able to send frames in order to keep the link utilization 100% (i.e. keeping the pipe full)

(BW x Delay) product = 0.5 Mbps (2x32 msec) = 32 Kbits = 4 Kbyte

Hence SWS = 4 KB/1KB = 4

b) The sequence of events are as follows:

Time (msec)	*Event*	*Comments*
0	*F_1 is transmitted*	
16	*F_2 is transmitted*	*This frame is lost*
32	*F_3 is transmitted*	*This frame is received and buffered (but not acknowledged)*
48	*F_4 is transmitted*	*The sender window has closed after transmitting this frame* *This frame is lost*
64	*ACK_1 is received* *F_5 is transmitted*	*The RTT is 64 msec. The Sender window size opened up one frame ⇒ Sender send F_5 and stops.*
112	*Timeout of F_2 has expired. Sender send F_2 again*	*This is just an assumption. The timeout is set at the time the frame is transmitted. You could assume that the time out starts from the end of the transmission of the frame*
128	*Timeout of F_3 has expired. Sender send F_4 again*	
144	*Timeout of F_4 has expired. Sender send F_4 again*	
160	*Timeout of F_5*	

	has expired. Sender send F_5 again	
176	*Sender receives ACK_3* *Send frame F_6*	*Receiver has acknowledged both F_2 and F_3. Sender window Will open by two frames*
192	*Send frame F_7*	
208	*Sender receives ACK_5*	*Receiver has acknowledged both F_4 and F_5. Sender window Will open by two frames and so on…..*