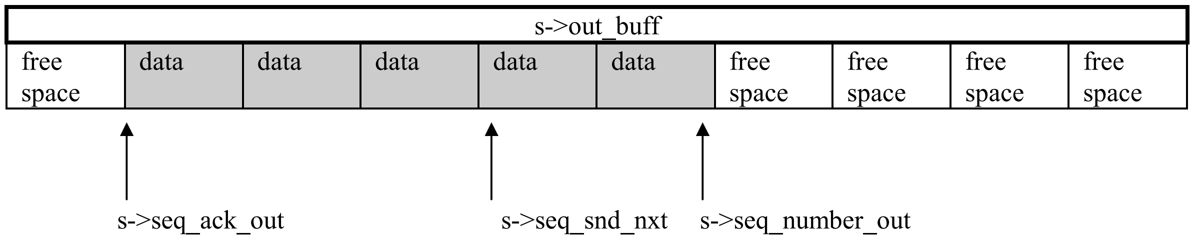
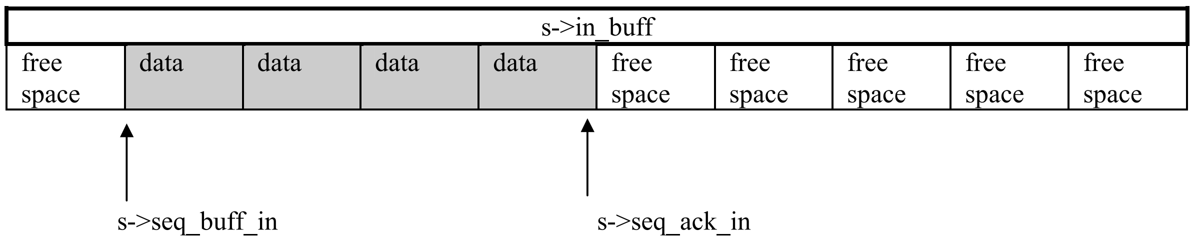
Socket Buffer Operation



seq_ack_out, seq_snd_next and seq_number_out are 32 bit numbers used by the TCP protocol. out_buff's size is defined by TCP_OUT_WINDOW_SIZE and must be equal to 2^n , where n=8-16. To determine the location inside the out_buff, the seq_ 32 bit numbers will be masked off using a mask that corresponds to TCP_OUT_WINDOW_SIZE defined as TCP_OUT_WINDOW_MASK.

s->seq_ack_out points to the beginning of the data in out_buff (window), it will also correspond to the last received acknowledgment number received from the other side of the TCP connection. When the other side of the connection returns an ack number greater than s->seq_ack_out, s->seq_ack_out is updated to this number and the free space size s->out_buff_free is updated to reflect the new amount of free space.

s->seq_snd_nxt points to the data that hasn't been sent yet, when it is this sockets turn to send data the TCP packet is filled with data starting at s->seq_snd_nxt and continuing up to the s->window_out maximum size as set by the other side of the connection or up until s->seq_number_out which is the last data byte available to transmit. After data has copied for transmission across the connection the s->seq_snd_nxt variable is updated to either s->seq_number_out or s->seq_snd_nxt+s->window_out.



s->seq_buff_in and s->seq_ack_in are 32 bit numbers used by the TCP protocol. in_buff's size is defined by TCP_IN_WINDOW_SIZE and and must be equal to 2^n , where n=8-16. To determine the location inside the in_buff, the seq_ 32 bit numbers will be masked off using a mask that corresponds to TCP_IN_WINDOW_SIZE defined as TCP_IN_WINDOW_MASK. Also TCP_IN_WINDOW_SIZE is advertised as the maximum window size when the SYN is sent.

s->seq_buff_in points to the beginning of the data to be read by the socket to be passed up to the application. When the application reads from the socket it will begin reading at s->seq_buff_in and continue to read until the application's buffer is full or until s->seq_ack_in. After the data is copied to the application's buffer s->seq_buff_in is updated either to s->seq_ack_in or to s->seq_buff + size of the applications buffer, also s->window_in is updated to reflect the free space in s->in_buff.

s->seq_ack_in point to the last byte in the s->in_buff. When a packet is received by the TCP software and is determined to be for this socket and corresponds to s->seq_ack_in+1, the data is copied to s->in_buff starting at s->seq_ack_in+1 and continuing until s->seq_buff_in or there is no more data in the incoming packet. s->seq_ack_in is also sent in every TCP packet to acknowledge data received.