Assignment 4, CPSC 441, Winter 2021, Lachlan Moore

When a timeout occurs the sending thread can only ‘send’ more segments if the queue is not equal to the window size given. While the receiving thread can only remove segments from the queue if they have been ACK’d. Because I use a concurrent queue the timeout function copies the queue into a list and retransmits all of those segments. The sender cannot add more to the queue until the receiver has made room by getting the next expected ACK.

If the sending thread sends a segment while the timer is retransmitting both will send and because an event like this can only happen when the queue is not full as segments are not removed until ACK’d. The segment sent by the sender thread would be the next one after the total queue that had been retransmitted.

In order to make sure that no two timers are created helper methods for starting and stopping the timer are made and are synchronized. This makes sure that only one thread can access the method to create a timer at a time, thus making sure that only one timer is running at once. Because the timer can only be started after an ACK or a send the timer should be stopped before another can be started.