Computer Science 441: Assignment 5

Name: Jack Yang UCID: 30062393

When a timeout occurs, only the segments that haven't been ACKed yet are retransmitted. Since a concurrent linked queue was used, there won't be any problems with concurrency with adding and removing from the queue. If the sending thread is adding the to the queue at the same time as the retransmission, then either the new segment is retransmitted right after being sent by the sending thread or not. Either way there's no harm in that. If the receiving thread is removed from the queue at the same time as the retransmission, then the segment being removed was already ACKed and confirmed to be valid, so it will either be retransmitted or not, both are safe because duplicate segments will be ignored at the server.

If a new segment is transmitted by the sending thread while the timer task is executing, nothing major would happen. As explained by the above, if the segment is added to the queue at the same time as retransmission is happening, then either the packet is sent again right after being sent the first time and the duplicate packet will be ignored at the server. Or the packet is added after the retransmission index is past the newest packet added therefore it won't be sent until the next retransmission happens.

If the receiving thread stops the timer task, then the ACK has already been received and processed, therefore nothing major happens. Even if the timer task is being started at the same time the start_timer() method is synchronized and in order to reach the start_timer method, the queue has to be read from first and the queue is also synchronized.