CPSC 441 Assignment 4

Name: Haohu Shen

UCID: 30063099

Tutorial: 04

What happens in the implementation if a timeout occurs at the same time that an ACK arrives from the server. Is there going to be a race condition in your program in that case?

Suppose the current sequence number in the client side is X, the client sent the packet with this sequence number to the server. If the client receives an ACK packet from the server, then there should be 2 cases:

Case 1: The ACK packet carries the sequence number of the next expected segment at the server, which is X+1, and the sequence number stored in the server side has been increased by 1.

Thus if a timeout occurs at the same time, the client will resend a packet with the old sequence number X, the server will receive it, but will ignore to update its sequence number after checking, then the server will send back the packet with the sequence number X+1 again to the client.

Since the client has received the correct ACK packet from the server, its sequence number has been updated to X+1, thus it will only receive the ACK packet with the sequence number X+2, thus the packet sent by the server which carries X+1 will be received and ignored after checking in the while loop.

Case 2: The ACK packet carries the sequence number is not X+1, then the client will receive it and ignore it after checking in the while loop and wait to receive the next packet. Thus if a timeout occurs at the same time, the client will send a packet with the sequence number X again, the server will receive it, and update its sequence number after checking, then send back the correct ACK packet with the sequence number X+1 to the client.

There is no race condition in both cases, since the ftp segment the timer task used to resend is a deep copy of the original segment and the procedure of the deep copy has been done before the initialization of the timer task.