

# CPE 464 – Stop and Wait Design Requirements Document

See Canvas for due date

**You may work on this in groups (up to 3 people). You only need to turn in one assignment for the group. All group members must be present for all work (except the final write-up). If you work in a group, you must provide a hardcopy of a picture of your group with some flow diagrams in the picture... and you MUST be smiling since its network related<sup>1</sup>!**

**Remember – You are NOT going to implement this program. We are focusing on the design of this program. You will use what you learn to help write program #3.**

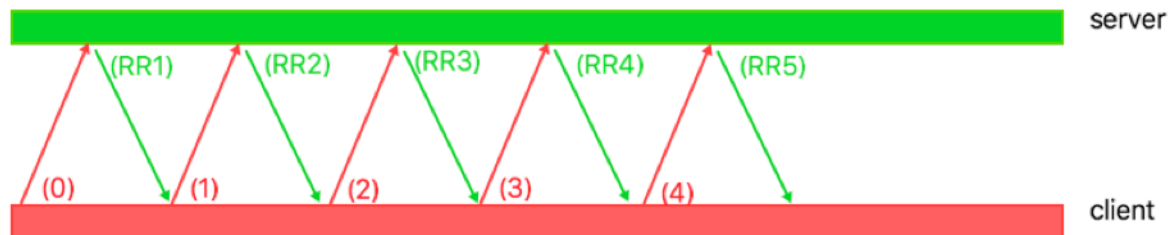
Note - Not all of these cases may be relevant for your design. These are just cases that need to be considered. If the case cannot happen or is handled some other way, just state that and if necessary explain why. Remember – design is a process not a direct journey. There are no correct answers, only some that work and some that don't. And yes, there are better designs and worse designs and that is part of the learning process.

## PART I – Design questions

- 1) Very High level: Draw a packet flow diagram of the following scenarios (use a diagram similar to the ones done in class – labeled arrows between rcopy and server, include setting a timer (i.e. calling select()) in your diagram.) and discuss what implications this has on your server/client:

- a. No packets lost

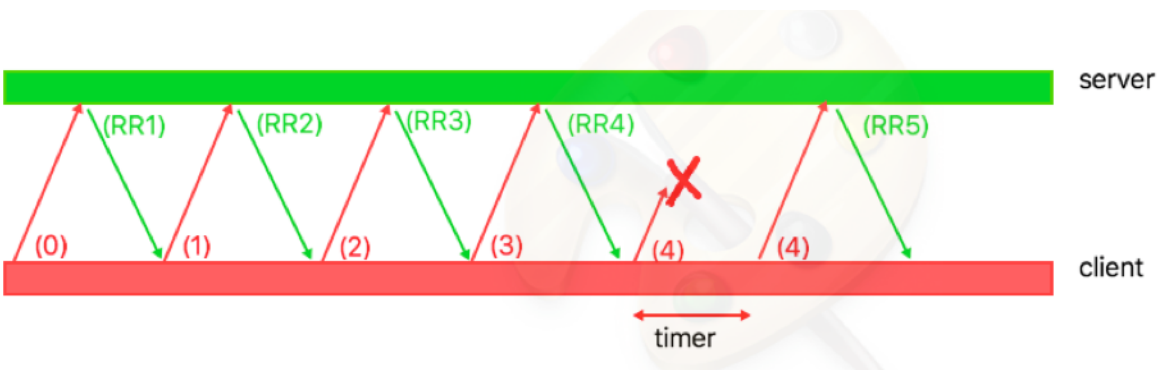
*No packet lost indicates a successful transmission and that means for every single packet that server receives from packet, it will send back a RR to indicate that it is ready to receive the next packet.*



- b. Data packet lost

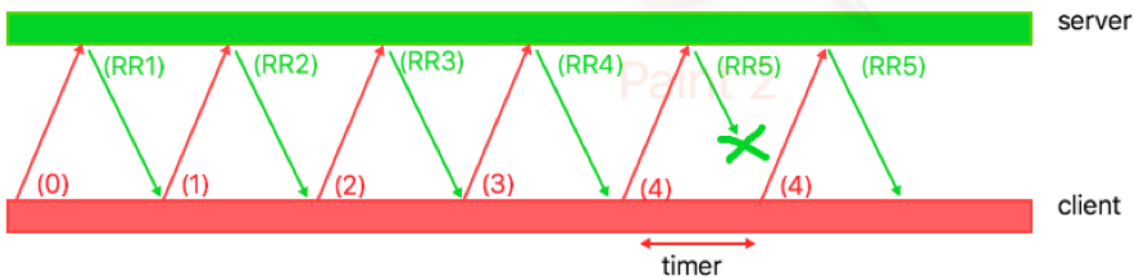
*If packet is lost, this means it must stop and wait. The only way for it to determine if a packet is lost before it reaches its destination is the client set a timer once it sends out a packet. If the client does not receive the ACK packet before timer goes out, then that's considered a lost packet. Therefore, the client would just send that same packet again.*

<sup>1</sup> Worthy pictures will find a spot on my office door. If you don't want your picture on my office door, please put a note on the picture.



c. ACK lost

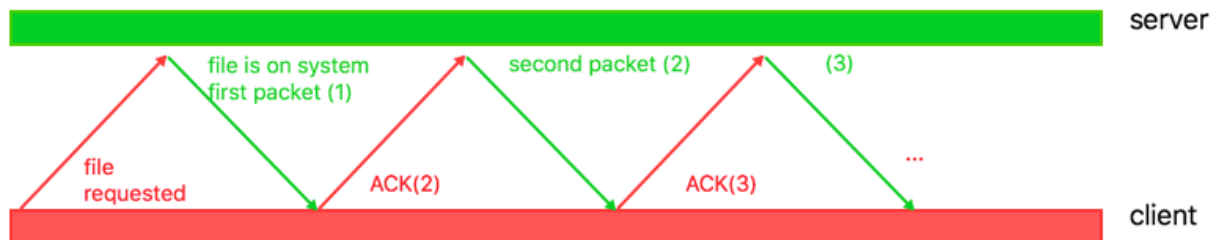
ACK lost means the server has received the packet from the client but the client does not know about this. This is still the same as the previous case, which means the client would just resend the packet again but the server would just not care about that packet but rather just care about sending back the ACK so that the client would know to send the next packet.



- 2) Draw a packet flow diagram for the following scenarios for the **filename** exchange or connection establishment exchange. End each diagram with the ACK for the second data packet being received.

a. No packets lost

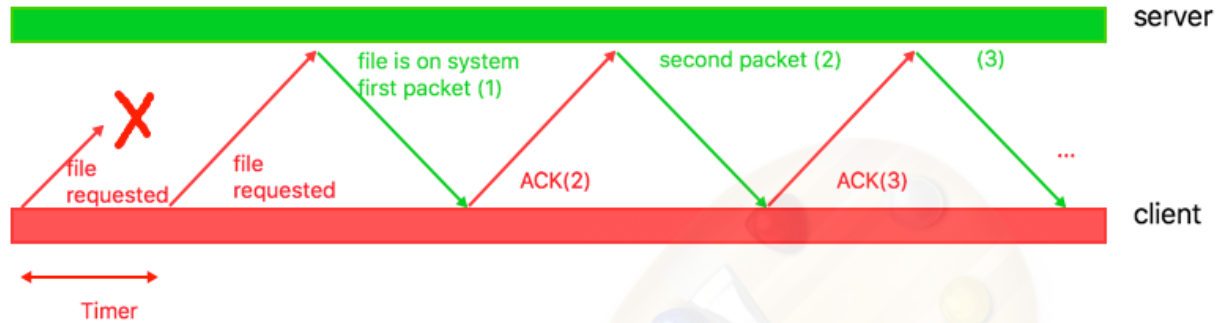
Well, if no packet is lost, then it's just the same way as number 1a. Client will just send the initial packet that indicate the filename and waiting for the first packet back from the server back. This packet is to indicate the file is on the server. After that, the client will tell the server that it has received an ACK from server to indicate that it's ready to gradually accept data. Every data packet sent, it will just return an ACK back to indicate that it's ready to accept the next data packet until server finishes sending all data that was asked based on the buffer size.



b. First packet sent by rcopy (filename or establishment packet) is lost

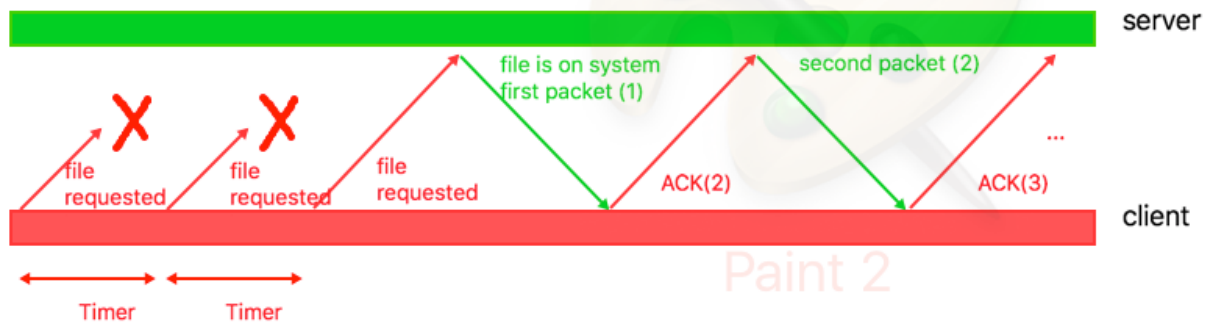
If filename or establishment packet is lost, that means the timer runs out before it receives a packet from server back that indicate file is on system. Then it will resend that packet

one more time and the rest is similar to 2a, where it will just gradually accept data.



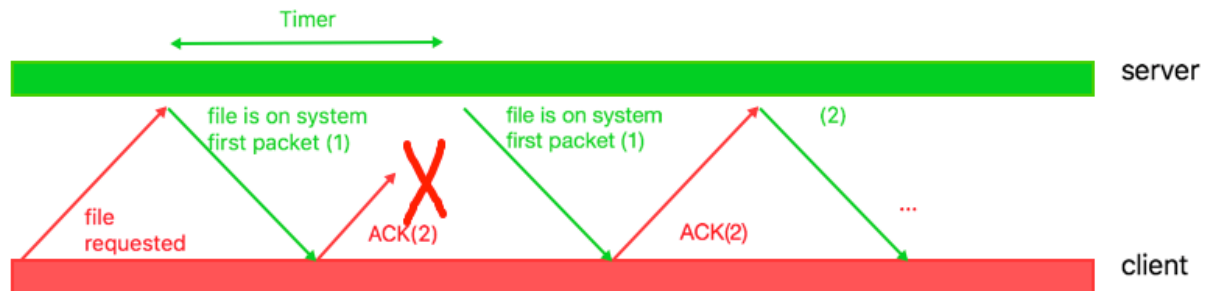
- c. First two packets sent by rcopy are lost.

This is the same as 2b, except it will just resend the first packet twice, instead of once. Then afterward, it would just gradually accept data packets from server and send ACK packets back to confirm and tell that it's ready for next data packet.



- d. Second packet sent by rcopy is lost (this might be the ACK for the 1<sup>st</sup> data packet)

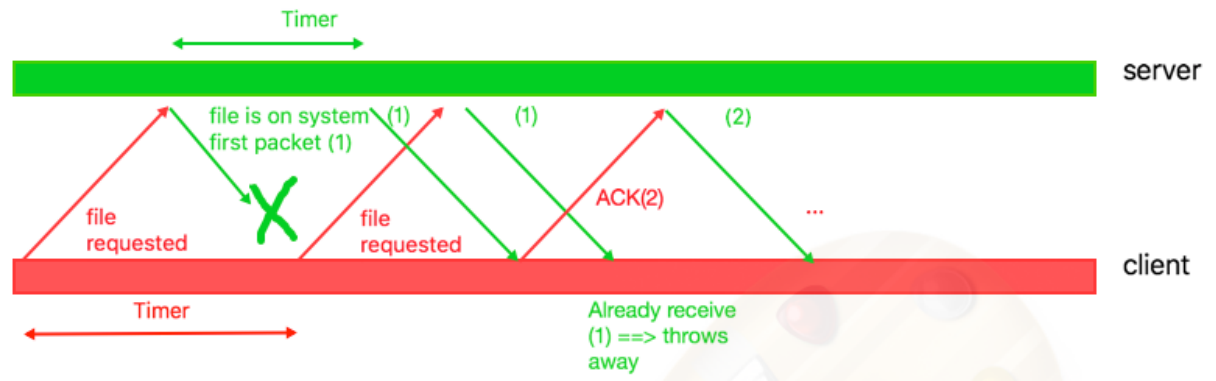
If second packet by rcopy is lost, and first packet wasn't lost, that means the first packet was successful. This means we are now on the data packet transferring. Since the ACK for first data packet is lost, there's no way for the server to know that the client has received its first data packet. The server, however, also has a timer to indicate that the packet has been lost, and therefore would just send the same packet again until it gets the ACK packet back from client. Then it would just gradually send the rest of data packets.



- e. First packet sent by the server is lost.

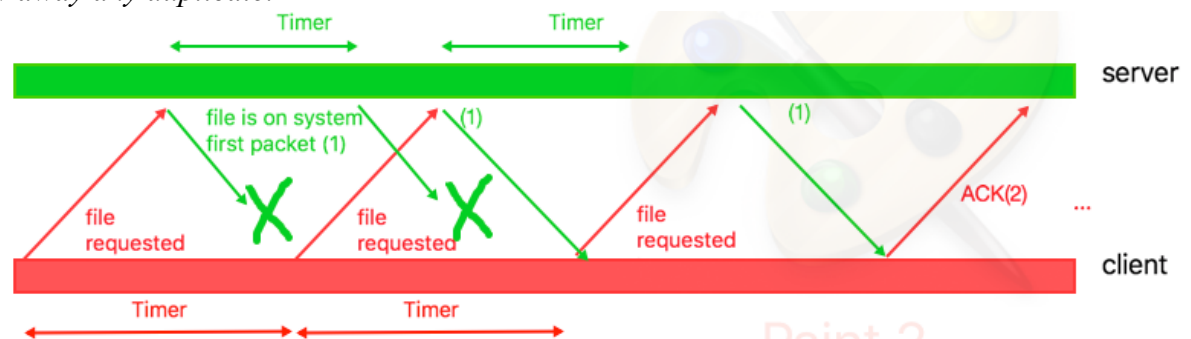
If first packet sent by the server is lost, that means the client does not fully know if file requested was on the server. Therefore, if timer of client runs out, it would just send file requested (packet 1) again to server and waiting for the acknowledgement back to indicate it's ready to accept data. However, the server also set a timer when it sent out their file on system first packet, but if that's expired, it would just send that packet again as well. And if the client requests for that packet again, then it would literally send back 2 packets but one would just be thrown away, and sent back with an ACK to indicate it's

now ready to receive data. In conclusion, the server would send back file is on system (first packet) 3 times to client but the first one is lost, and the second one would be from timer; and third one is from the second file request but that one is thrown away due to the fact that it's already received it from the second sent.



- f. First two packets sent by the server are lost.

This is quite similar to the previous case. If the first packet is lost, it went through the same process before it starts sending out data to client. However, in this case, the first and second packet is lost, meaning client would not receive it on the first two packets. This depends on how long the timer is set on each side, but going from previous example, this just means that the client will just accept the packet on the third try and would know throw away any duplicate.



- g. First data packet sent by the server is lost

With the first data packet sent by server is lost, then it just means it's the same as 2e. It would just have some piggy back ride and duplicate sent of packets that need to be thrown away on the receiver end.

- 3) Looking back at questions #1 and #2, list the possible packet scenario(s) that will cause select() to time out.

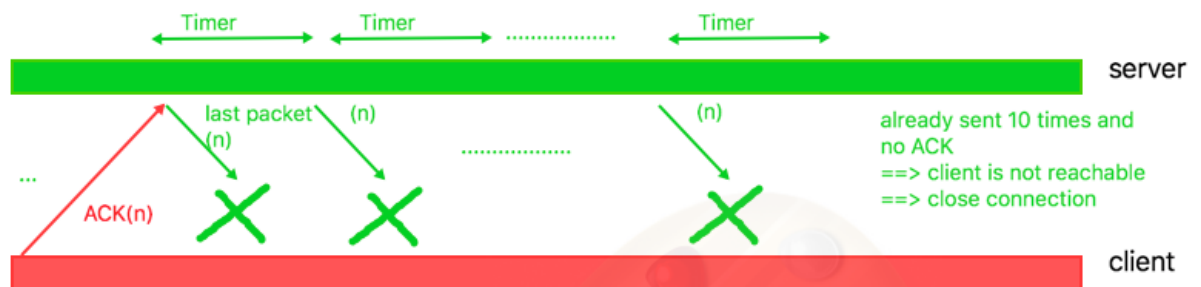
- 1) What would cause select to time out is if one side is terminated for whatever reason. That just means it will not return any data, and therefore would cause the timer to expires on the sender side.
- 2) The case from previous packet that causes select() to time out is if first file requested packet is lost. And also, the data packet is lost.
- 3) Server's timer would time out if the ACK packet is lost or client's side terminated after requesting the file.

4) Regarding the following possible scenarios for receiving data. Explain **how** this scenario can happen and list **what action** you will take with the data and how you will reply to the sender:

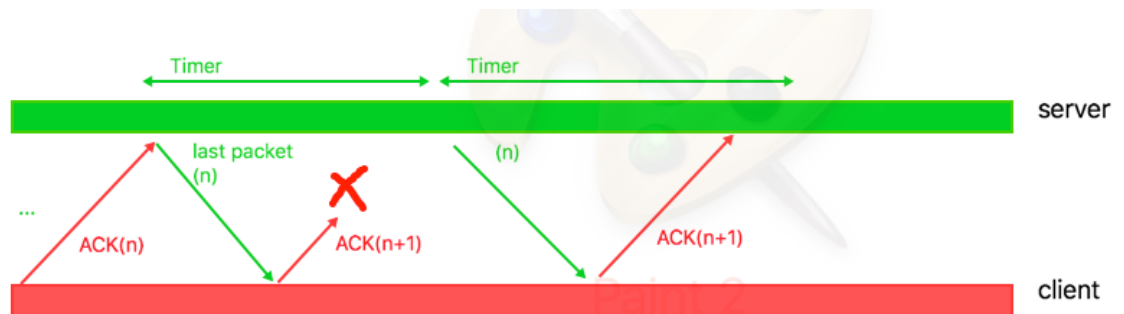
- Data sequence number is the one you expect  
*This just means everything's working well. So it will just send back an ACK to indicate it's ready to receive next packet.*
- Data sequence number is a duplicate of one you have already received  
*Then it would just throw away the duplicated one and not reset the timer. Since it's already sent the ACK back, it would just wait for the next expected packet.*
- The data packet is corrupted  
*The data packet is corrupted, then it would just throw it away and not send back an ACK. The server's timer side would time out then, and it would just resend the packet.*
- The ACK is corrupted  
*If ACK is corrupted, it would just be similar to 2b, then because the timer of server side would run out and it would just resend the same packet back. Then, it will send away the duplicated one, and then wait til timer to run out, and send back the new ACK packet.*

5) Give a packet flow diagram on how you will handle the last packet of the file.

- Last data packet is lost  
*When last data packet is lost, the timer of server would time out. It would time out again if the last data packet is lost again. However, when it reaches the 10th times, server would just indicate that client has terminated because its destination is not reachable.*



- ACK from rcopy for last data packet is lost.  
*If ACK from rcopy back to server is lost, and it's lost 10 times, that means it thinks server side is terminated. I think in this case it doesn't really matter because it's already received all the data packet it's needed, but it would just try to reestablish the connection to server if it wants to get any new data. Server would also keep sending packet 10 times, but client would just throw the 9 duplicate ones away. Eventually, both will think each side has terminated.*



- 6) **Make a copy of your answers to this assignment** ... you will not get this back in time to write the next program.

## **Part II – State diagrams**

Turn in a state diagram for rcopy and another for server (so two diagrams). These diagrams should start with the filename exchange and cover the sending/acking of the last packet.

**Make a copy of your packet flows and state diagrams before you turn them in...** you will not get them back in time to design and write the next program.

