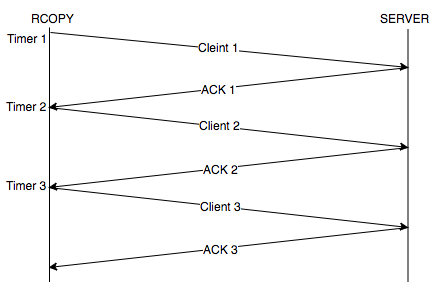
**Stop and Wait Program Design**

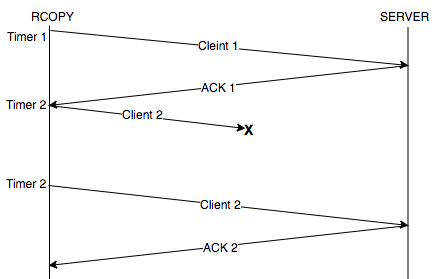
*Part 1: Design Questions*

1. Server/Client Packet Flow Diagrams
   1. No packets lost



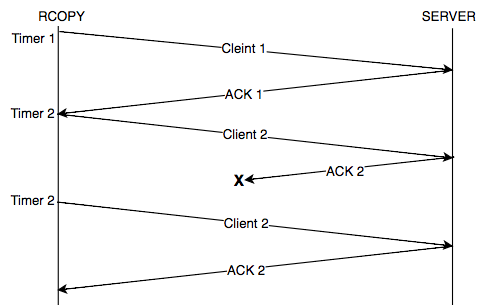
Select is called as each client packet is sent. Server ACKs each packet received. Client does not send another packet until the ACK is received.

* 1. Data packet lost



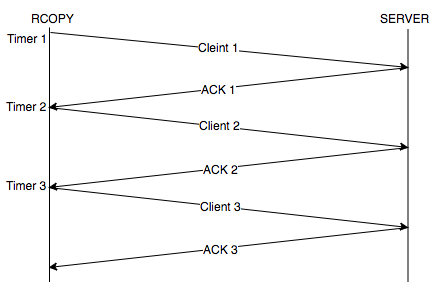
2nd packet is lost (either data is lost or corrupted). As such, the server will not ACK, and timer 2 will timeout – causing the client to send the same packet again. Upon successful transmission, the server ACKs the 2nd packet.

* 1. ACK lost



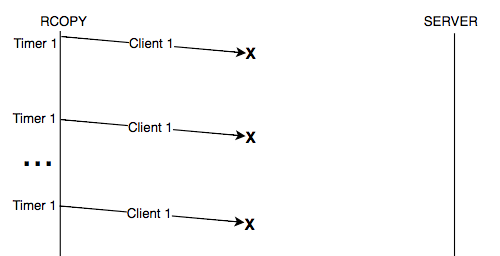
Server receives 2nd packet, but ACK 2 gets lost on the way back. Client’s Timer 2 times out, and sends the packet again. Server ignores this data and sends another ACK 2.

1. Filename Packet Flow Diagrams
   1. No packets lost



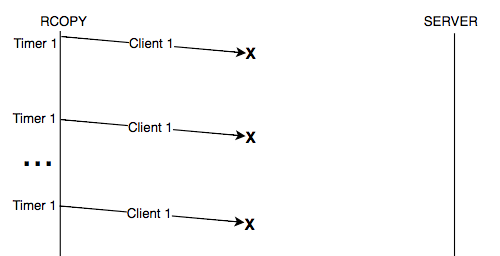
Filename established with First client packet. No packets are lost, so the data transmission continues smoothly.

* 1. First packet sent by rcopy is lost



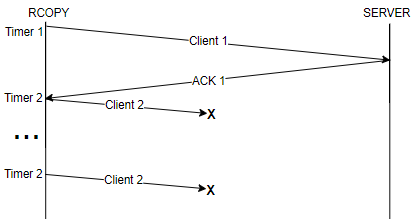
Client tries to send the first packet up to 10 times, and then infers that the server is unreachable. Client terminates.

* 1. First two packets sent by rcopy are lost



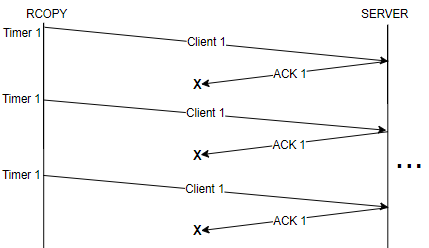
Client tries to send the first packet up to 10 times, and then infers that the server is unreachable. Client terminates. Same situation would happen for the second packet if the first packet has made it

* 1. Second packet sent by rcopy is lost



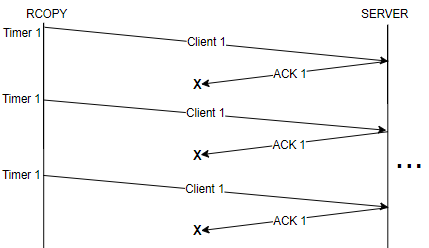
First packet makes it so server. Upon receiving ACK 1, the second packet tries to send and times out for up to 10 times. Client then terminates – assuming server is unreachable.

* 1. First packet sent by the server is lost



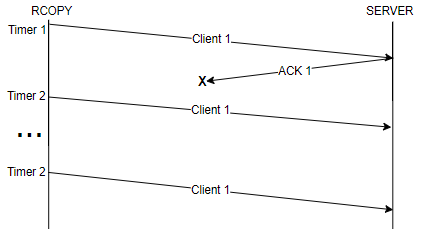
First ACK never makes it to the client. Client times out up to 10 times and then assumes the server is unreachable.

* 1. First two packets sent by the server are lost



First ACK never makes it to the client. Client times out up to 10 times and then assumes the server is unreachable. Same things happens if the first ACK makes it but the second ACK gets stuck

* 1. First data packet sent by the server is lost



First ACK never makes it. Client times out after 10 sends.

1. Select timeouts:
   1. Data packet loss leading to no ACKs
   2. ACK never makes it back to client
2. Receive Data Use Cases
   1. **Data sequence number is the one you expect**

Send back an ACK

* 1. **Data sequence number is a duplicate of one you have already received**

Toss the incoming message and ACK that sequence number again.

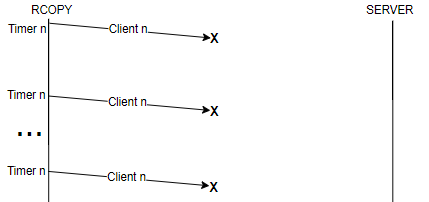
* 1. **The data packet is corrupted**

Toss the incoming message and do not ACK

* 1. **The ACK is corrupted**

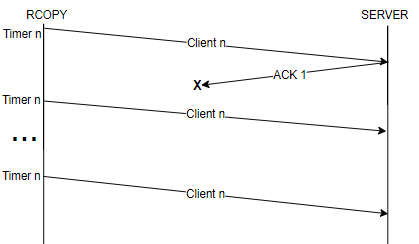
Send the sequence again.

1. Last Packet Flow Diagram
   1. Last data packet is lost



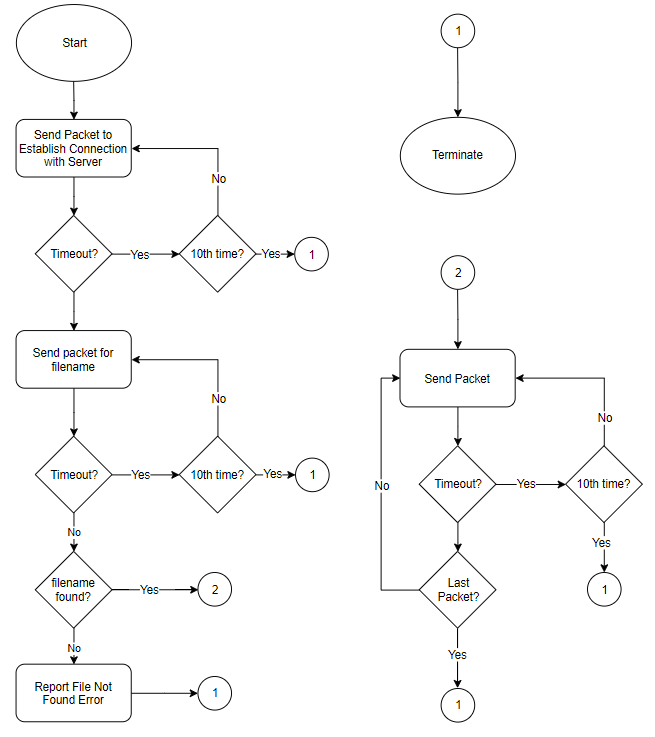
Last packet sends up to 10 times but never receives an ACK. Server is unreachable, so client terminates.

* 1. ACK from rcopy for last data packet is lost



*Part 2: State Diagram*

**Client: RCOPY –**



**Server –**

