This protocol thrives over UDP protocol to bring principles of reliable data transfer by constructing additional features in the middle layer over the UDP socket to transfer files between a sender and a receiver.

Events occuring at the SENDER side:

1. The given file is read and each line is made into a packet, with the line string as the payload.
2. Connection is setup with the reciever, using the reciever address and port number, by creating a socket, using mySocket() .
3. The file is sent in parts of 4096 bytes in the form of custom created packets.
4. When a packet is sent, a thread is created which will wait to recieve the Ack for this packet, thus implementing Multithreading.
5. If the packet is lost, its thread retransmits the packet and waits for an Ack again, in which case a duplicate packet may be recived by receiver which is handled by the sequence number.
6. When an Ack is recieved, the sender stops sending

Events occurring at the RECIEVER side:

1. The reciever sets up a a socket using mySocket() and binds to the user provided port number.
2. The reciver then waits to recieve packets from the sender.
3. If a packet is received , it is checked for errors and an Ack is sent to sender with the sequence numbered packet.
4. If a packet with sequence number that was earlier recieved is recieved again, it is acknowledged as a duplicate packet, and an Ack is sent as the sender could have lost the previous Ack.

Strategies to tackle various situations:

1. Packet Loss: The packet loss is handled by waiting for acks
2. Packet Corruption : The packets are included with the checksum field which is checked for against the on spot calculated checksum at the receiver side.
3. Duplicate acks: The duplicates are handled by the sequence number as sequences other than the anticipated ack sequences are discarded.

Assumptions in Application layer:

1. The file to be sent is present in the current working directory
2. The file to be received is always wanted in present directory of the receiver

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx Add here xxxxxxxxxxxxxxxxxxxxx

Assumptions in Network Layer:

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx Add here xxxxxxxxxxxxxxxxxxxxx

Additional Scenario:

1. The middle layer is designed such that it can change it’s policy of reliable data transfer into plain UDP accordingly if the user permits unreliability.