README FILE: Question 2

Author: Ayush Jain
BITS ID: 2017A7PS0093P

Execution Instructions:

1. In directory q2, write **make** in the terminal.
2. Open three terminals in the same directory. First execute ./server in one terminal. In the 2nd terminal please execute ./relay. Then execute ./client in 3rd terminal

Description of Files in Directory.

1. Client.c : code for client.
2. Server.c: code for server
3. Packet.h: Header file for packet.h
4. Selective_repeat.h: Header file for server.c, client.c, relay.c
5. Makefile
6. Input.txt: A large txt file, to be transferred by client to server.
7. Input1.txt: A smaller subset of input.txt
8. Clientlog.txt: Log of terminal when ./client is executed
9. Serverlog.txt: Log of terminal when ./server is executed
10. Relaylog.txt: Log for both the relays combined when ./relay is executed.
11. Client.o: Binary for direct execution using ./client
12. Server.o : Binary for direct execution using ./server
13. Server.o: Binary for direct execution using ./server
14. Output.txt: file at server after transmission

Important #defines.

PACKET.h

    1. #define PACKET_SIZE : modify this for changing the size of payload of packet (ignoring the other fields). This specifies the number of bytes of data transferred.
    2. #define timeout: Amount of time after the packet timeouts if ACK is not received.

Client.c

1. #define PORT_RELAY1: refers to the 1st relay's port to send data to.
2. #define PORT_RELAY2: refers to the 2nd relay's port to send data to.
3. #define WINDOW_SIZE: sliding window size at client, presently set to 10

Server.c

1. #define PORT: refers to server port which accepts data from relays.
2. #define BUFFER_SIZE: This is a sliding window buffer, which slides when the head of the window got the data. It keeps on sliding until it gets data sequentially. When the data is not in order and the head packet of the window is not available, it does not slide.

Relay.c

1. #define PDR: Packet drop rate. Must be between 0 and 1. 0.1 means 10% drop rate.
2. #define MAX_ACK_DELAY: Time in milliseconds. Ack packets will be delayed by a random millisecond time between 0 to MAX_ACK_DELAY.
3. #define PORT_SERVER: Server's port.
4. #define PORT_RELAY1: Relay1's port to which client sends data to. It further can send data to the server's port.
5. #define PORT_RELAY2: Relay2's port to which client sends data to. It further can send data to the server's port.

Execution status: Able to transfer a big file of approximately 150000 bytes from client to server using relay nodes. Tested correctly with different values of packet drop rates, ACK delay values.

KEY DESIGN DECISIONS

1. Packets are dropped at the relay side with PDR probability. A message is shown “OOPS, Packet dropped at relay” for convenience of tracking the execution of the code.
2. Timeout is implemented for the whole window and not for individual packets. Only unacked packets are retransmitted as per the selective repeat protocol. A message “Timeout occurred: Retransmitting packets” is written on the terminal for help in tracking the execution.
3. At client side a window is maintained of 10 packets(can be modified by changing #define window_size value). Even packets are sent to relay1 and odd packets are sent to relay2. Whenever we receive an ack for some packet(through any channel), we slide the window if possible and send new packets. For example, if I send 10 packets initially and I get ACK for packet 1, I slide my window and send packet 11. But if ACK for 1 is not received, and ACK of 2 is received we do not slide. We set the flag of the window item for Packet2 to mark that its ACK is received. Now, after some time if we receive ACK for 1, we can slide the window two times and send two new packets.
4. At Relay, we can have packets from server as well as client. We can know if the packet is an ACK or data by looking into the packet structure (isDataNotACK will be true if its data).
5. If it's an ACK, we send it to the server. Here we also add some delay for the ACK packets as per the documentation.
6. If its data, we can either drop it with PDR probability (“OOPS, packet is dropped at relay” is seen on terminal”), or we can send it to the server.
7. Server implements a sliding window buffer with functionality similar to what was described in point 1. It takes care of all the edge cases.