USER MANUAL

for

COMPUTER NETWORKS ASSIGNMENT

Prepared by: -

Jackson Stephan (B200743CS)

Sayooj Geo Ninan (B200058CS)

Course: CS3006D COMPUTER NETWORKS

Date: 13/04/2023

Contents

1. Problem Statement

It is required to create an application to simulate a sender and a receiver. Using UDP sockets, the sender must send 100 ms and 150 ms interval packets repeatedly to the receiver.

The receiver must receive and process the received packets using 4 threads: -
1st thread: Receiving the packets sent by the sender.
2nd thread: Process the packets of type 1.
3rd thread: Process the packets of type 2.
4th thread: Prints the number of packets of each type received at every 300 ms interval.

2. Steps of execution

Step 1: Make a directory and keep the client and server code in separate C files.

Step 2: Open 2 terminal windows and go to the directory which contains the C program files for the client and the server.

Step 3: Compile the client and server C program files in separate terminal windows using the following commands: -
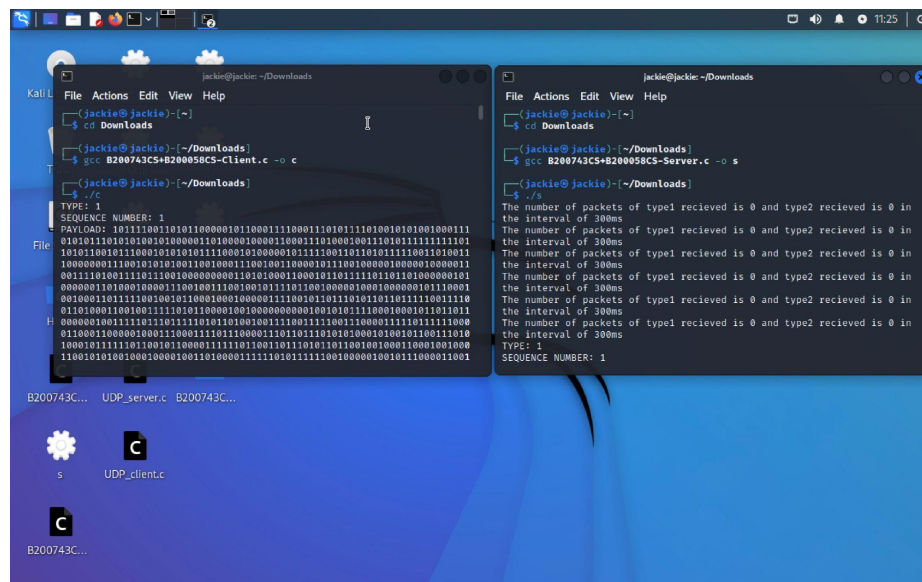
gcc B200743CS+B200058CS-Client.c -o c
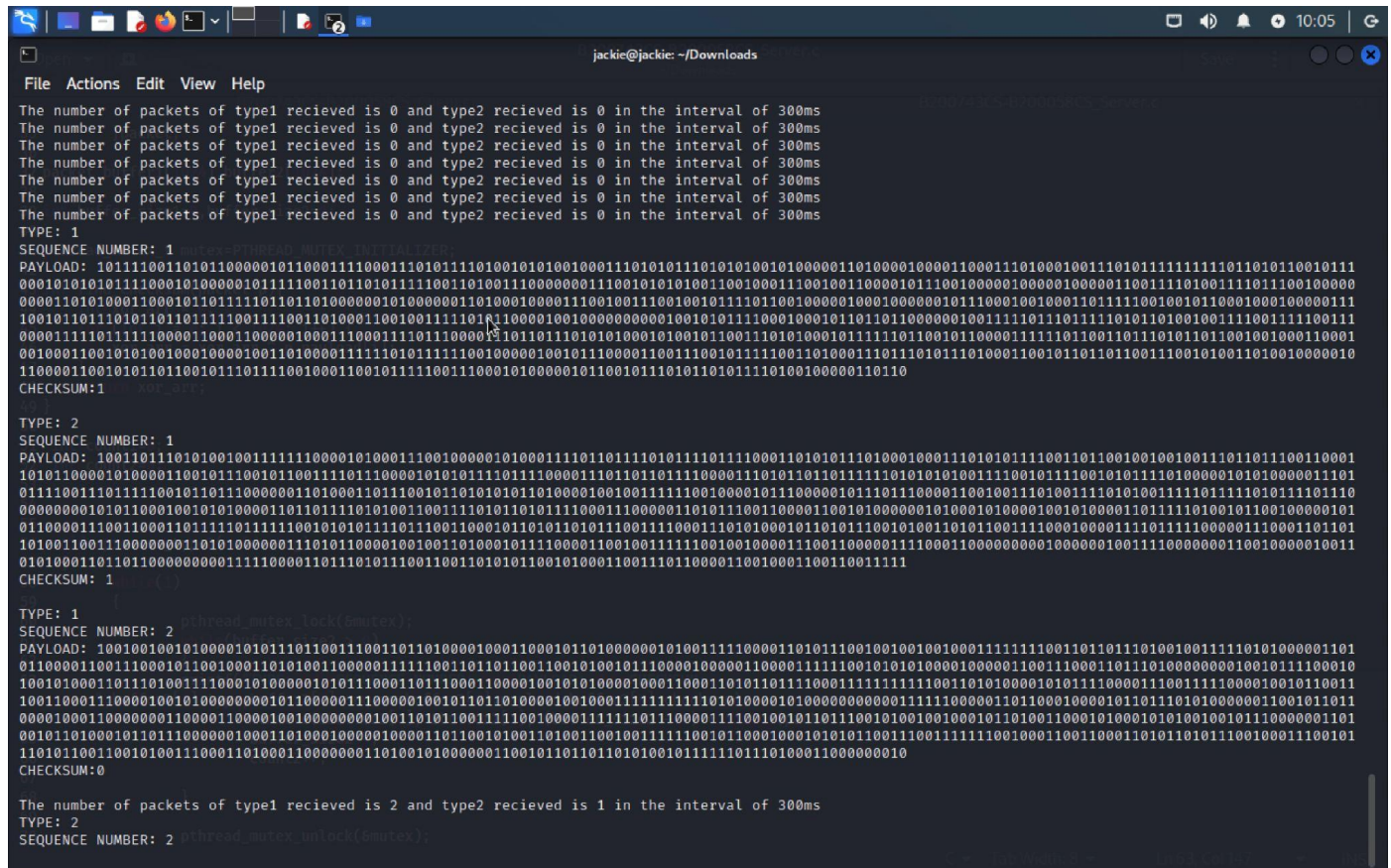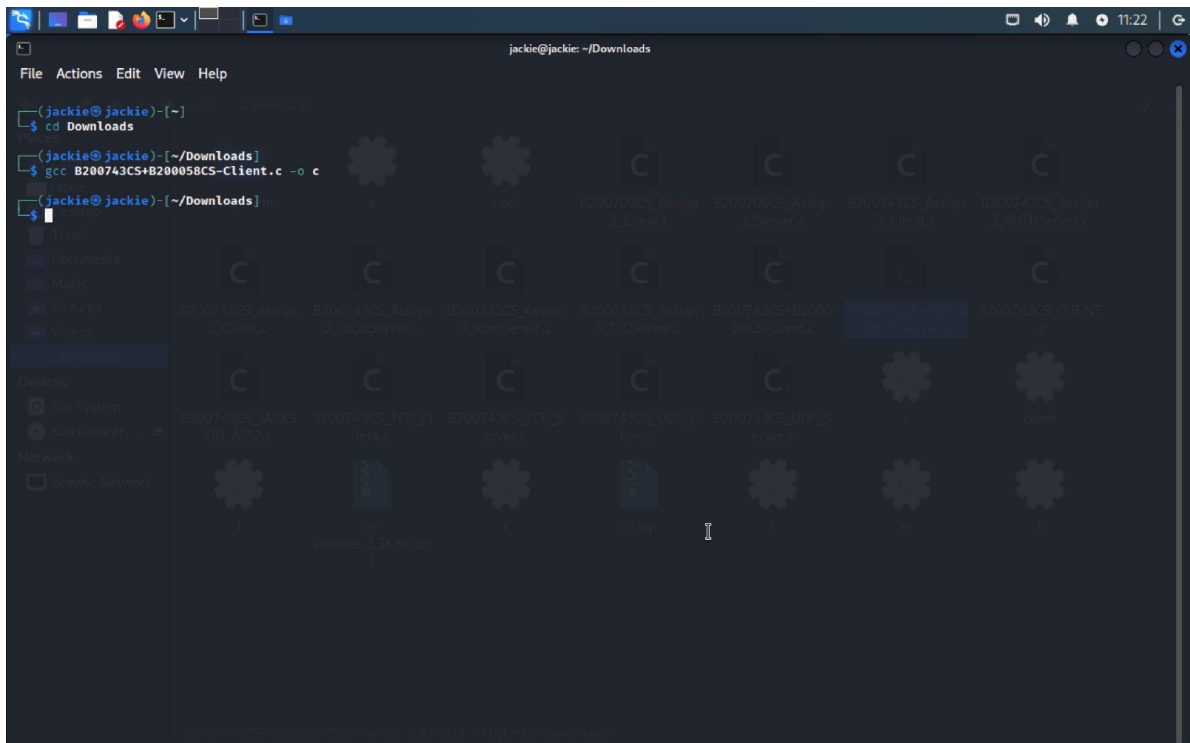
gcc B200743CS+B200058CS-Server.c -o s
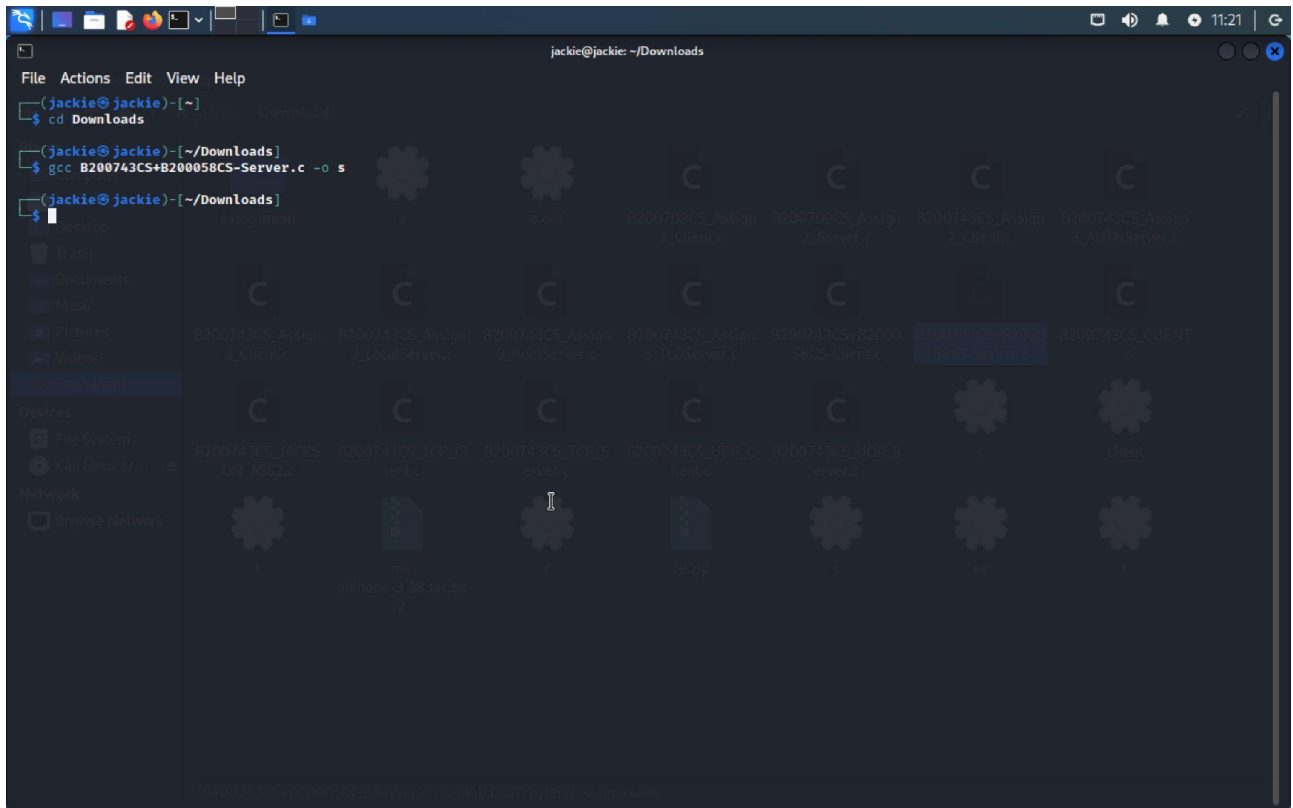
Step 4: Run the compiled server code first, followed by execution of the client code using the commands ./s and ./c respectively.

3. Output Specification

3.1) Server file

The server file is expected to print the details of the received packet which includes packet type,sequence number,payload and checksum and after every 300 ms, it prints the number of packets effectively sent between the client and server for both packet types.

3.2) Client file

The client file is expected to print the details of the packets sent which includes packet type,sequence number,payload and checksum.

```
jackie@jackie: ~/Downloads
File  Actions  Edit  View  Help

┌──(jackie㉿jackie)-[~/Downloads]
└─$ ./c
TYPE: 1
SEQUENCE NUMBER: 1
PAYLOAD: 10111100110101100000101100011110001110101111010010101001000111010101110101010010100000110100001000011000111010001001110101111111110101010110010111
00010101010101111000101000001011111001101101011110011011010011100000011100101010100110010001110010011000010111001000001000001000001100111101010011110011100100000
00001101010001100010110111110110110100000010100000011010001000011100100110011001001011110011001000010001000000101110010010001001101111100100101100010001000100000111
10010110111010110111111001111110010101101000110010011111010110000010000000000000010010101111001000010010110110110000001001111101110111111101010010011011001111100111
00001111110111111100001100010010000010000110011110101100011110110111101010100010100101011101010010001111111011100101011001011100010110100101101100100110010010001000110001
001001100101010010010000100010000011111110101011111100100000100101100001001011100101011100110101101111001101010001110110101011101010001001000110010011010010000010
110000110010010110110011101111100100010011001010111100111000010000010110010110111010110101011110100100000000110110
CHECKSUM: 1

TYPE: 2
SEQUENCE NUMBER: 1
PAYLOAD: 100110111010100100111111110000101000011001000001010001111011101111010111101111000110101011101000100011101010111100110110010010010010011101101111001100110001
10101100001010000110010111001011001110110111000010100101110111011000011101101101111110101010100111001011110010101111010000010101000000011101
01111001110111100010110111100011011000010110100101010101001101011101101110000101011011110010101001010101111001100001011110001110101000000000101111011000000110100
0000000010010110010101010001011010111101010010010111100100111001011111000111001111000001110101011100011100010001110101010000100110100010101001010000101000000010110101001011111
01100001110011000110111110111111001010101011101100110000101101011010101100111100011101010001011010111001010001010010110101110010010011010110010111000010000111101
1010001001101110100010001111010110001000100101010001011110001100100111111001001000011100110010000011100010000000000100000001001110000000011001000000100011
0101000110110110000000000111101101101101110011001101010100101000110011101101100001100100010011001011111
CHECKSUM: 1

TYPE: 1
SEQUENCE NUMBER: 2
PAYLOAD: 100100100101000010101101100111001101101100000100011000101110100000001010011110000011010111001001001001000111111110010110111011101001001111110101010000001101
011000011001110010010011010001001101001111000011111001101101100100100010010101010001010000011011110000001000110110001101110101101010001010011100100011000100110
10010100011011110100111100010100000101011100110111001101011011000011100000010101011100001111111110011010100000010101110011001111001001111001110010110011
1001100011100000100101000000001011000000011100000100101101101000001001000111111111111010100010010000000000011110000001101100100000101011110110110000100110101011
00001000110000000010010001000010010101111100010010011110110001010011001110001010010010100100100010100101010001010001001011100000001101
001011011000010110110000010001011010000100010000110011001010100010111110110000001001001011111001010100010111001001010101001010010010111001001010001000101100011100101
1101011001100101001110001100011000000011010010100000001100101101101010100101111011010001100000010
CHECKSUM: 0

TYPE: 2
SEQUENCE NUMBER: 2
PAYLOAD: 1100011111110001001010000000011111111000110011000010010011010101010100100110111111011011110100000010011000011100011101110101001001000110011111101011
000101111111001010010111101000010101100010110101111000001010110110010101011111011111011111111101001011011000010100101101011101110011011101111010000
0101100010111100001000101011001111010010010010100111001011001011000110101010010000101110011011100110101000011001111111101100110011100011011
110101100101101110100010100011000010101001001011000011001110101000010100010110001011010010110110001010101001011110000001110111001001000101001101011011101011011101101110100010010100011110111010
001011000111111001011001010100111010110101011100010001011001101010101000011010111110110110111100110011101111100011001110111011110111010100010110100010001111111101100001111111011000010001111100
```
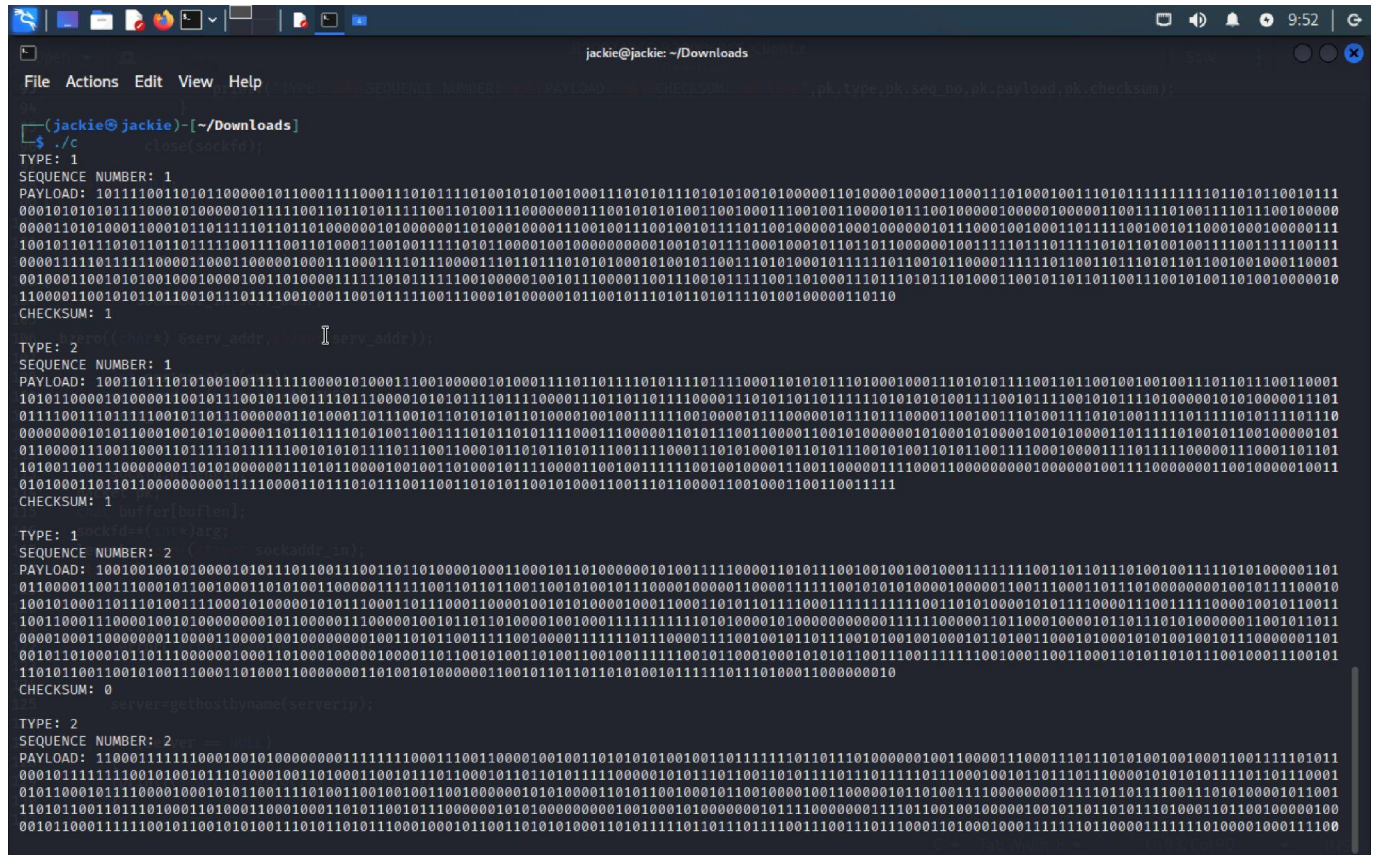
4. Conclusion

The client successfully sends packets of 100ms and 150ms using 2 threads. The server receives all the sent packets using a single thread, which also checks the checksum for any transmission errors. Parallely, the other 2 threads in the server end print the details of the packets. The 4th thread in the server side, displays a count of the number of type 1 and type 2 packets effectively sent between the client and the server during an interval of 300ms. The concepts used for successfully simulating the process are **socket programming**,**multithreading and shared variables**.

5. Authors

1. Jackson Stephan
   jackson_b200743cs@nitc.ac.in
   Ph.no. 8301919896

2. Sayooj Geo Ninan
   sayooj_b200058cs@nitc.ac.in
   Ph.no. 8714032402