

# **ASSIGNMENT 4 – CONCURRENCY CONTROL AND RELIABLE DATA TRANSFER WITH UDP**

**MD AZHARULLAH SHARIFF  
B130727CS  
S6, B.TECH, CSE**

## **Programs :**

- 1) Sender – 4\_B130727s.c
- 2) Receiver – 4\_B130727r.c

## **1) 4\_B130727s.c :**

### **Client :**

First, the client creates the sockets and allots the requires values to it. Then it shows the user a menu and asks the user to enter his choice. Based on the user choice, the following actions take place:

1) Choice = 1 => Stop and wait transmission with all scenarios(without concurrency – one direction) :

Here, a counter is maintained in the loop and each message is assigned values automatically as : seqnum = counter and acknum = -1. The payload is taken as input from user and assigned accordingly. Then it is sent the server. The

server, if successfully receives the packet, sets the acknum to the corresponding packet number and send back the acknum to the client.

2) Choice = 2 => Stop and wait transmission with all scenarios (with concurrency – simultaneous transmission in both directions) :

Here, a counter is maintained in the loop and each message is assigned values automatically as : seqnum = counter and acknum = -1. The payload is taken as input from user and assigned accordingly. Then it is sent to the server. The server, if successfully receives the packet, sets the acknum to the corresponding packet number and send back the acknum to the client. Also, the server sends back another packet to the client. The client then sets back the acknum for that particular packet and sends it back to the server. The server can get packets and send packets from/to multiple clients simultaneously with the help of the select mechanism to handle the concurrency issues.

3) Choice = 3 => Selective repeat with all scenarios (without concurrency- one direction) :

Here, a file of data is maintained and is divided into chunks based on the window size. First, the first chunk is sent to the server and the server analyzes the chunk and sends back the acknum for that particular chunk. Then the client moves the select window forward and sends the next chunk to the server. If, at any point of time, the client doesn't get the acknum for any chunk within a specified time, then the chunk is again sent to the server.

4) Choice = 4 => Selective repeat with all scenarios (with concurrency - simultaneous transmission in both directions) :

Here, a file of data is maintained and is divided into chunks based on the window size. First, the first chunk is sent to the server and the server analyzes the chunk and sends back the acknum for that particular chunk. Then the client moves the select window forward and sends the next chunk to the server. If, at any point of time, the client doesn't get the acknum for any chunk within a specified time, then the chunk is again sent to the server. The server can get chunks and send chunks from/to multiple clients simultaneously with the help of the select mechanism to handle the concurrency issues.

The choice is sent to the server also.

**2) 4\_B130727r.c :**

## **Server :**

First, the server creates the sockets and assigns values to them appropriately. Then it receives the choice number from the client and then does the appropriate actions as follows :

1) Choice = 1 => Stop and wait transmission with all scenarios (without concurrency – one direction) :

Here, the server waits till it receives a packet from the client. The server, if successfully receives the packet, sets the acknum to the corresponding packet number and sends back the acknum to the client.

2) Choice = 2 => Stop and wait transmission with all scenarios (with concurrency – simultaneous transmission in both directions) :

Here, the server waits till it receives a packet from the client. The server, if successfully receives the packet, sets the acknum to the corresponding packet number and sends back the acknum to the client. Also, the server sends back another packet to the client. The client then sets back the acknum for that particular packet and sends it back to the server. The server can get packets and send packets from/to multiple clients simultaneously with the help of the select mechanism to handle the concurrency issues.

3) Choice = 3 => Selective repeat with all scenarios (without concurrency-one direction) :

Here, the server waits till it receives a chunk of data from the client. The server, if successfully receives the chunk, sets the acknum for the chunk appropriately and sends back the acknum to the client.

4) Choice = 4 => Selective repeat with all scenarios (with concurrency - simultaneous transmission in both directions) :

Here, the server waits till it receives a chunk of data from the client. The server, if successfully receives the chunk, sets the acknum for the chunk appropriately and sends back the acknum to the client. The server also can get chunks and send chunks from/to multiple clients simultaneously with the help of the select mechanism to handle the concurrency issues.