**Assignment 3 TCP-RUDP**

**Computer Network**

**Program: MScIT Sem-2**

**Group ID : 28**

**Student Name Student ID**

Dev Adnani 202212012

Saif Saiyed 202212083

# 202212012

1: Time Control

Client.c

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <stdio.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

int noOfPck(int size){

int req = size/100;

double reqD = size/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,int size,

struct control\_pkt\* c\_pkt){

int pkts = noOfPck(size);

printf("\nPackets in handshake: %d\n",pkts);

c\_pkt->end\_sr\_no = pkts;

c\_pkt->sr\_no = 0;

c\_pkt->no\_of\_pkt = pkts;

c\_pkt->start\_sr\_no=1;

c\_pkt->checksum = createSumControl(c\_pkt);

//ack

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

while(ack->ack != 1){

if(sendto(sock\_fd,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.\n");

return -1;

}

printf("\nSent handshake.");

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

}

return 1;

}

int comunicate(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length){

//msg

struct data\_pkt dataSend;

struct data\_pkt \* data1 = &dataSend;

memset(data1->msg,0,100);

printf("\nEnter msg: ");

fgets(data1->msg,100,stdin);

data1->sr\_no=1;

data1->checkSum = createSum(data1);

if(sendto(sock\_fd,data1,sizeof(\*data1),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.");

return -1;

}

printf("\nSent: %s",data1->msg);

struct ack\_pkt dataRecv;

struct ack\_pkt \* ack = &dataRecv;

ack->ack=-1;

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

if(ack->ack==-1){

printf("\nNo ack recved");

}

else{

printf("\nAck recved");

}

return 1;

}

int comunicatePkt(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,char \* msg,int sr\_no){

struct data\_pkt dataSend;

struct data\_pkt \* data = &dataSend;

memset(data->msg,0,100);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

int i=0;

while(\*msg != '\0'){

data->msg[i]=\*msg;

msg++;i++;

}

data->sr\_no = sr\_no;

data->checkSum = createSum(data);

if(sendto(sock\_fd,data,sizeof(\*data),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server");

return -1;

}

printf("\nSent packet %d : %s ",sr\_no,data->msg);

printf("\nchecksum: %ld",data->checkSum);

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

printf("\nack: %d",ack->ack);

return ack->ack;

}

int main(){

int sock\_fd;

struct sockaddr\_in server\_addr;

int sock\_length = sizeof(server\_addr);

sock\_fd = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(sock\_fd < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nSocket created.");

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_family = AF\_INET;

char msg[200];

int size = sizeof(msg)/sizeof(msg[0]);

struct control\_pkt pkt;

int handshakeResult = shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

}

printf("\nSuccefull handshake\n");

printf("\nNo of packets: %d.",pkt.no\_of\_pkt);

char \*pkt\_datas[2];

char \* f = "hello";

char \* s = "world";

printf("here");

pkt\_datas[0]=f;

pkt\_datas[1]=s;

for(int i=0;i<pkt.no\_of\_pkt;i++){

int result = comunicatePkt(sock\_fd,server\_addr,sock\_length,pkt\_datas[i],i);

while(result == -1){

result = comunicatePkt(sock\_fd,server\_addr,sock\_length,pkt\_datas[i],i);

}

}

return 0;

}

Client Screenshot :



Server.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

void setZero(struct control\_pkt \* pkt){

pkt->sr\_no=0;

pkt->no\_of\_pkt=0;

pkt->checksum=0;

pkt->end\_sr\_no=0;

pkt->start\_sr\_no=0;

}

int noOfPck(char \* msg){

int len = strlen(msg);

int req = len/100;

double reqD = len/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,

struct control\_pkt \* c\_pkt){

setZero(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=0;

ack->sr\_no=0;

int valid = 0;

while(valid == 0){

if(recvfrom(socket\_desc,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

printf("\nRecved control pkt\n");

valid = checkSumControl(c\_pkt);

ack->ack=1;

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending akc to client.");

return -1;

}

}

printf("\nSent control pkt ack.\n");

return 1;

}

int comunicate(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=0;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

}

int comunicatePkt(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,int sr\_no){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nchecksum: %ld",createSum(data1));

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

if(data1->sr\_no != sr\_no) return -1;

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=-1;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

return ack->ack;

}

int main(){

int socket\_desc;

struct sockaddr\_in server\_addr,client\_addr;

int client\_addr\_size = sizeof(client\_addr);

socket\_desc = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(socket\_desc < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nCreated socket.");

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

//binding to port and ip

if(bind(socket\_desc,(struct sockaddr \*)&server\_addr,sizeof(server\_addr))<0){

printf("\nBinding error");

return -1;

}

printf("\nBinding of socket.");

printf("\nNow listening...");

struct control\_pkt pkt;

int handshakeResult = shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

while(handshakeResult==-1){

return -1;

}

printf("\nPackets : %d\n",pkt.no\_of\_pkt);

for(int i=0;i<pkt.no\_of\_pkt;i++){

int result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

while(result == -1){

result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

}

}

return 1;

}

Server Screenshot :



Q2 : TCP

Client.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <arpa/inet.h>

#define SIZE 1024

#define IP "127.0.0.1"

#define PORT 5050

void send\_file(FILE \* fp,int sockfd){

int n;

char data[SIZE]={0};

while(fgets(data,SIZE,fp) != NULL){

printf("\n%s",data);

if(send(sockfd,data,sizeof(data),0)==-1){

printf("\nError while sending file.");

return;

}

memset(data,0,SIZE);

}

}

int main(){

int sockfd,clientfd;

struct sockaddr\_in server\_addr;

FILE \*fp;

char \*filename = "test.txt";

sockfd = socket(AF\_INET,SOCK\_STREAM,0);

if(sockfd < 0){

printf("\nError while creating socket");

return -1;

}

printf("\nCreated to socket");

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr(IP);

clientfd = connect(sockfd,(struct sockaddr \*)&server\_addr,sizeof(server\_addr));

if(clientfd == -1){

printf("\nError while connecting");

return -1;

}

printf("\nConnected to server");

fp=fopen(filename,"r");

if(fp==NULL){

printf("\nError while reading file");

return -1;

}

send\_file(fp,sockfd);

printf("\nFile data sent");

return 0;

}

Client Screenshot :



Server.c

#include<unistd.h>

#include<stdio.h>

#include<sys/socket.h>

#include<stdlib.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 5050

#define SIZE 1024

void write\_file(int sockfd){

int n;

FILE \*fp;

char \* filename = "get.txt";

char buffer[SIZE];

fp = fopen(filename,"w");

while(1){

n = recv(sockfd,buffer,SIZE,0);

if(n<=0){break;return;}

printf("\n%s",buffer);

fprintf(fp,"%s",buffer);

memset(buffer,0,1024);

}

return;

}

int main(){

int server\_fd,new\_socket,valRead;

struct sockaddr\_in address;

int opt = 1;

int addrlen = sizeof(address);

char buffer[1024];

char \*hello = "Hello from server";

if((server\_fd = socket(AF\_INET,SOCK\_STREAM,0))==0){

printf("\nSocket creation error");

return -1;

}

printf("\nSocket Created");

if(setsockopt(server\_fd,SOL\_SOCKET,SO\_REUSEADDR | SO\_REUSEPORT,&opt,sizeof(opt))){

printf("\nSetsockopt");

return -1;

}

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(PORT);

if(bind(server\_fd,(struct sockaddr \*)&address , sizeof(address))<0){

printf("\nBinding Error");

return -1;

}

printf("\nSocket Binded");

if(listen(server\_fd,3)<0){

nnections

printf("\nListening Error");

return -1;

}

printf("\nSocket Listening");

if((new\_socket = accept(server\_fd,(struct sockaddr \*)&address, (socklen\_t \*)&addrlen))<0){

epting clients connection

printf("\nAccepting Error");

return -1;

}

printf("\nSocket Accepted request\n");

write\_file(new\_socket);

printf("Data saved");

return 0;

}

Screenshot : 

Q3 : UDP

Client.c

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <stdio.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

int noOfPck(int size){

int req = size/100;

double reqD = size/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,int size,

struct control\_pkt\* c\_pkt){

int pkts = noOfPck(size);

printf("\nPackets in handshake: %d\n",pkts);

c\_pkt->end\_sr\_no = pkts;

c\_pkt->sr\_no = 0;

c\_pkt->no\_of\_pkt = pkts;

c\_pkt->start\_sr\_no=1;

c\_pkt->checksum = createSumControl(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

while(ack->ack != 1){

if(sendto(sock\_fd,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.\n");

return -1;

}

printf("\nSent handshake.");

//time delay check

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

}

return 1;

}

int comunicate(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length){

struct data\_pkt dataSend;

struct data\_pkt \* data1 = &dataSend;

memset(data1->msg,0,100);

printf("\nEnter msg: ");

fgets(data1->msg,100,stdin);

data1->sr\_no=1;

data1->checkSum = createSum(data1);

if(sendto(sock\_fd,data1,sizeof(\*data1),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.");

return -1;

}

printf("\nSent: %s",data1->msg);

struct ack\_pkt dataRecv;

struct ack\_pkt \* ack = &dataRecv;

ack->ack=-1;

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

if(ack->ack==-1){

printf("\nNo ack recved");

}

else{

printf("\nAck recved");

}

;

return 1;

}

int comunicatePkt(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,char \* msg,int sr\_no){

struct data\_pkt dataSend;

struct data\_pkt \* data = &dataSend;

memset(data->msg,0,100);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

int i=0;

while(\*msg != '\0'){

data->msg[i]=\*msg;

msg++;i++;

}

data->sr\_no = sr\_no;

data->checkSum = createSum(data);

if(sendto(sock\_fd,data,sizeof(\*data),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server");

return -1;

}

printf("\nSent packet %d : %s ",sr\_no,data->msg);

printf("\nchecksum: %ld",data->checkSum);

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

printf("\nack: %d",ack->ack);

return ack->ack;

}

int main(){

int sock\_fd;

struct sockaddr\_in server\_addr;

int sock\_length = sizeof(server\_addr);

sock\_fd = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(sock\_fd < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nSocket created.");

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_family = AF\_INET;

char msg[100];

int size = sizeof(msg)/sizeof(msg[0]);

memset(msg,0,100);

struct control\_pkt pkt;

int handshakeResult = shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

}

printf("\nSuccefull handshake\n");

printf("\nNo of packets: %d.",pkt.no\_of\_pkt);

char \* filename="test.txt";

FILE \*fp = fopen(filename,"r");

if(fp == NULL) {

printf("\nError while reading file");

}

int i=0;

while(fgets(msg,100,fp) !=NULL){

int result = comunicatePkt(sock\_fd,server\_addr,sock\_length,msg,i);

while(result == -1){

result = comunicatePkt(sock\_fd,server\_addr,sock\_length,msg,i);

}

i++;

}

return 0;

}

Client Screenshot



Server.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

void setZero(struct control\_pkt \* pkt){

pkt->sr\_no=0;

pkt->no\_of\_pkt=0;

pkt->checksum=0;

pkt->end\_sr\_no=0;

pkt->start\_sr\_no=0;

}

int noOfPck(char \* msg){

int len = strlen(msg);

int req = len/100;

double reqD = len/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,

struct control\_pkt \* c\_pkt){

setZero(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=0;

ack->sr\_no=0;

int valid = 0;

while(valid == 0){

if(recvfrom(socket\_desc,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

printf("\nRecved control pkt\n");

valid = checkSumControl(c\_pkt);

ack->ack=1;

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending akc to client.");

return -1;

}

}

printf("\nSent control pkt ack.\n");

return 1;

}

int comunicate(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=0;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

}

int comunicatePkt(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,int sr\_no){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nchecksum: %ld",createSum(data1));

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

if(data1->sr\_no != sr\_no) return -1;

ack->ack=1;

char \* filename = "get.txt";

FILE \*fp = fopen(filename,"a");

fprintf(fp,"%s",data1->msg);

printf("\nmessage: %s",data1->msg);

printf("\nData pkt is valid\n");

}

else{

ack->ack=-1;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

return ack->ack;

}

int main(){

int socket\_desc;

struct sockaddr\_in server\_addr,client\_addr;

int client\_addr\_size = sizeof(client\_addr);

socket\_desc = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(socket\_desc < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nCreated socket.");

p

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

if(bind(socket\_desc,(struct sockaddr \*)&server\_addr,sizeof(server\_addr))<0){

printf("\nBinding error");

return -1;

}

printf("\nBinding of socket.");

printf("\nNow listening...");

struct control\_pkt pkt;

int handshakeResult = shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

return -1;

}

printf("\nPackets : %d\n",pkt.no\_of\_pkt);

for(int i=0;i<3;i++){

int result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

while(result == -1){

result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

}

}

return 1;

}

Server Screenshot :



# 202212083

1: Time Control

Client.c

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <stdio.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

int noOfPck(int size){

int req = size/100;

double reqD = size/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,int size,

struct control\_pkt\* c\_pkt){

int pkts = noOfPck(size);

printf("\nPackets in handshake: %d\n",pkts);

c\_pkt->end\_sr\_no = pkts;

c\_pkt->sr\_no = 0;

c\_pkt->no\_of\_pkt = pkts;

c\_pkt->start\_sr\_no=1;

c\_pkt->checksum = createSumControl(c\_pkt);

//ack

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

while(ack->ack != 1){

if(sendto(sock\_fd,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.\n");

return -1;

}

printf("\nSent handshake.");

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

}

return 1;

}

int comunicate(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length){

//msg

struct data\_pkt dataSend;

struct data\_pkt \* data1 = &dataSend;

memset(data1->msg,0,100);

printf("\nEnter msg: ");

fgets(data1->msg,100,stdin);

data1->sr\_no=1;

data1->checkSum = createSum(data1);

if(sendto(sock\_fd,data1,sizeof(\*data1),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.");

return -1;

}

printf("\nSent: %s",data1->msg);

struct ack\_pkt dataRecv;

struct ack\_pkt \* ack = &dataRecv;

ack->ack=-1;

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

if(ack->ack==-1){

printf("\nNo ack recved");

}

else{

printf("\nAck recved");

}

return 1;

}

int comunicatePkt(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,char \* msg,int sr\_no){

struct data\_pkt dataSend;

struct data\_pkt \* data = &dataSend;

memset(data->msg,0,100);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

int i=0;

while(\*msg != '\0'){

data->msg[i]=\*msg;

msg++;i++;

}

data->sr\_no = sr\_no;

data->checkSum = createSum(data);

if(sendto(sock\_fd,data,sizeof(\*data),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server");

return -1;

}

printf("\nSent packet %d : %s ",sr\_no,data->msg);

printf("\nchecksum: %ld",data->checkSum);

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

printf("\nack: %d",ack->ack);

return ack->ack;

}

int main(){

int sock\_fd;

struct sockaddr\_in server\_addr;

int sock\_length = sizeof(server\_addr);

sock\_fd = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(sock\_fd < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nSocket created.");

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_family = AF\_INET;

char msg[200];

int size = sizeof(msg)/sizeof(msg[0]);

struct control\_pkt pkt;

int handshakeResult = shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

}

printf("\nSuccefull handshake\n");

printf("\nNo of packets: %d.",pkt.no\_of\_pkt);

char \*pkt\_datas[2];

char \* f = "hello";

char \* s = "world";

printf("here");

pkt\_datas[0]=f;

pkt\_datas[1]=s;

for(int i=0;i<pkt.no\_of\_pkt;i++){

int result = comunicatePkt(sock\_fd,server\_addr,sock\_length,pkt\_datas[i],i);

while(result == -1){

result = comunicatePkt(sock\_fd,server\_addr,sock\_length,pkt\_datas[i],i);

}

}

return 0;

}

Output:



Server.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

void setZero(struct control\_pkt \* pkt){

pkt->sr\_no=0;

pkt->no\_of\_pkt=0;

pkt->checksum=0;

pkt->end\_sr\_no=0;

pkt->start\_sr\_no=0;

}

int noOfPck(char \* msg){

int len = strlen(msg);

int req = len/100;

double reqD = len/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,

struct control\_pkt \* c\_pkt){

setZero(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=0;

ack->sr\_no=0;

int valid = 0;

while(valid == 0){

if(recvfrom(socket\_desc,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

printf("\nRecved control pkt\n");

valid = checkSumControl(c\_pkt);

ack->ack=1;

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending akc to client.");

return -1;

}

}

printf("\nSent control pkt ack.\n");

return 1;

}

int comunicate(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=0;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

}

int comunicatePkt(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,int sr\_no){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nchecksum: %ld",createSum(data1));

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

if(data1->sr\_no != sr\_no) return -1;

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=-1;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

return ack->ack;

}

int main(){

int socket\_desc;

struct sockaddr\_in server\_addr,client\_addr;

int client\_addr\_size = sizeof(client\_addr);

socket\_desc = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(socket\_desc < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nCreated socket.");

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

//binding to port and ip

if(bind(socket\_desc,(struct sockaddr \*)&server\_addr,sizeof(server\_addr))<0){

printf("\nBinding error");

return -1;

}

printf("\nBinding of socket.");

printf("\nNow listening...");

struct control\_pkt pkt;

int handshakeResult = shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

while(handshakeResult==-1){

return -1;

}

printf("\nPackets : %d\n",pkt.no\_of\_pkt);

for(int i=0;i<pkt.no\_of\_pkt;i++){

int result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

while(result == -1){

result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

}

}

return 1;

}

Server Screenshot :



Q2 : TCP

Client.c

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <arpa/inet.h>

#define SIZE 1024

#define IP "127.0.0.1"

#define PORT 5050

void send\_file(FILE \* fp,int sockfd){

int n;

char data[SIZE]={0};

while(fgets(data,SIZE,fp) != NULL){

printf("\n%s",data);

if(send(sockfd,data,sizeof(data),0)==-1){

printf("\nError while sending file.");

return;

}

memset(data,0,SIZE);

}

}

int main(){

int sockfd,clientfd;

struct sockaddr\_in server\_addr;

FILE \*fp;

char \*filename = "test.txt";

sockfd = socket(AF\_INET,SOCK\_STREAM,0);

if(sockfd < 0){

printf("\nError while creating socket");

return -1;

}

printf("\nCreated to socket");

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr(IP);

clientfd = connect(sockfd,(struct sockaddr \*)&server\_addr,sizeof(server\_addr));

if(clientfd == -1){

printf("\nError while connecting");

return -1;

}

printf("\nConnected to server");

fp=fopen(filename,"r");

if(fp==NULL){

printf("\nError while reading file");

return -1;

}

send\_file(fp,sockfd);

printf("\nFile data sent");

return 0;

}

Client Screenshot :



Server.c

#include<unistd.h>

#include<stdio.h>

#include<sys/socket.h>

#include<stdlib.h>

#include<netinet/in.h>

#include<string.h>

#define PORT 5050

#define SIZE 1024

void write\_file(int sockfd){

int n;

FILE \*fp;

char \* filename = "get.txt";

char buffer[SIZE];

fp = fopen(filename,"w");

while(1){

n = recv(sockfd,buffer,SIZE,0);

if(n<=0){break;return;}

printf("\n%s",buffer);

fprintf(fp,"%s",buffer);

memset(buffer,0,1024);

}

return;

}

int main(){

int server\_fd,new\_socket,valRead;

struct sockaddr\_in address;

int opt = 1;

int addrlen = sizeof(address);

char buffer[1024];

char \*hello = "Hello from server";

if((server\_fd = socket(AF\_INET,SOCK\_STREAM,0))==0){

printf("\nSocket creation error");

return -1;

}

printf("\nSocket Created");

if(setsockopt(server\_fd,SOL\_SOCKET,SO\_REUSEADDR | SO\_REUSEPORT,&opt,sizeof(opt))){

printf("\nSetsockopt");

return -1;

}

address.sin\_family = AF\_INET;

address.sin\_addr.s\_addr = INADDR\_ANY;

address.sin\_port = htons(PORT);

if(bind(server\_fd,(struct sockaddr \*)&address , sizeof(address))<0){

printf("\nBinding Error");

return -1;

}

printf("\nSocket Binded");

if(listen(server\_fd,3)<0){

nnections

printf("\nListening Error");

return -1;

}

printf("\nSocket Listening");

if((new\_socket = accept(server\_fd,(struct sockaddr \*)&address, (socklen\_t \*)&addrlen))<0){

epting clients connection

printf("\nAccepting Error");

return -1;

}

printf("\nSocket Accepted request\n");

write\_file(new\_socket);

printf("Data saved");

return 0;

}

Screenshot :



Q3 : UDP

Client.c

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <stdio.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

int noOfPck(int size){

int req = size/100;

double reqD = size/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,int size,

struct control\_pkt\* c\_pkt){

int pkts = noOfPck(size);

printf("\nPackets in handshake: %d\n",pkts);

c\_pkt->end\_sr\_no = pkts;

c\_pkt->sr\_no = 0;

c\_pkt->no\_of\_pkt = pkts;

c\_pkt->start\_sr\_no=1;

c\_pkt->checksum = createSumControl(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

while(ack->ack != 1){

if(sendto(sock\_fd,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.\n");

return -1;

}

printf("\nSent handshake.");

//time delay check

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

}

return 1;

}

int comunicate(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length){

struct data\_pkt dataSend;

struct data\_pkt \* data1 = &dataSend;

memset(data1->msg,0,100);

printf("\nEnter msg: ");

fgets(data1->msg,100,stdin);

data1->sr\_no=1;

data1->checkSum = createSum(data1);

if(sendto(sock\_fd,data1,sizeof(\*data1),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server.");

return -1;

}

printf("\nSent: %s",data1->msg);

struct ack\_pkt dataRecv;

struct ack\_pkt \* ack = &dataRecv;

ack->ack=-1;

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

if(ack->ack==-1){

printf("\nNo ack recved");

}

else{

printf("\nAck recved");

}

;

return 1;

}

int comunicatePkt(int sock\_fd,struct sockaddr\_in server\_addr,int sock\_length,char \* msg,int sr\_no){

struct data\_pkt dataSend;

struct data\_pkt \* data = &dataSend;

memset(data->msg,0,100);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=-1;

int i=0;

while(\*msg != '\0'){

data->msg[i]=\*msg;

msg++;i++;

}

data->sr\_no = sr\_no;

data->checkSum = createSum(data);

if(sendto(sock\_fd,data,sizeof(\*data),0,

(struct sockaddr \*)&server\_addr,sock\_length)<0){

printf("\nError while sending message to server");

return -1;

}

printf("\nSent packet %d : %s ",sr\_no,data->msg);

printf("\nchecksum: %ld",data->checkSum);

struct timeval t;

t.tv\_sec=5;

fd\_set socks;

FD\_ZERO(&socks);

FD\_SET(sock\_fd,&socks);

if(select(sock\_fd+1,&socks,NULL,NULL,&t) &&

recvfrom(sock\_fd,ack,sizeof(\*ack),0,

(struct sockaddr \*)&server\_addr,&sock\_length)<0){

printf("\nError in delay recv function.");

}

printf("\nack: %d",ack->ack);

return ack->ack;

}

int main(){

int sock\_fd;

struct sockaddr\_in server\_addr;

int sock\_length = sizeof(server\_addr);

sock\_fd = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(sock\_fd < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nSocket created.");

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_family = AF\_INET;

char msg[100];

int size = sizeof(msg)/sizeof(msg[0]);

memset(msg,0,100);

struct control\_pkt pkt;

int handshakeResult = shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(sock\_fd,server\_addr,sock\_length,size,&pkt);

}

printf("\nSuccefull handshake\n");

printf("\nNo of packets: %d.",pkt.no\_of\_pkt);

char \* filename="test.txt";

FILE \*fp = fopen(filename,"r");

if(fp == NULL) {

printf("\nError while reading file");

}

int i=0;

while(fgets(msg,100,fp) !=NULL){

int result = comunicatePkt(sock\_fd,server\_addr,sock\_length,msg,i);

while(result == -1){

result = comunicatePkt(sock\_fd,server\_addr,sock\_length,msg,i);

}

i++;

}

return 0;

}

Client Screenshot



Server.c

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#define PORT 5100

struct control\_pkt{

int sr\_no;

int no\_of\_pkt;

int start\_sr\_no;

int end\_sr\_no;

int checksum;

};

struct data\_pkt{

int sr\_no;

char msg[100];

long checkSum;

};

struct ack\_pkt{

int ack;

int sr\_no;

long checkSum;

};

void setZero(struct control\_pkt \* pkt){

pkt->sr\_no=0;

pkt->no\_of\_pkt=0;

pkt->checksum=0;

pkt->end\_sr\_no=0;

pkt->start\_sr\_no=0;

}

int noOfPck(char \* msg){

int len = strlen(msg);

int req = len/100;

double reqD = len/(double)100;

if(reqD > req) req++;

return req;

}

long createSumControl(struct control\_pkt \* c\_pkt){

long sum=0;

sum+=c\_pkt->end\_sr\_no;

sum+=c\_pkt->no\_of\_pkt;

sum+=c\_pkt->sr\_no;

sum+=c\_pkt->start\_sr\_no;

return sum;

}

int checkSumControl(struct control\_pkt \* c\_pkt){

long check = createSumControl(c\_pkt);

if(c\_pkt->checksum == check) return 1;

return 0;

}

long createSumACK(struct ack\_pkt\* ack){

return ack->ack+ack->sr\_no;

}

int checkSumACK(struct ack\_pkt \* ack){

long ackSum = createSumACK(ack);

if(ackSum == ack->checkSum) return 1;

return 0;

}

long createSum(struct data\_pkt\* data){

long msgSum=0;

for(int i=0;i<100;i+=2){

msgSum += ((data->msg[i]<<8) + data->msg[i+1]);

}

msgSum += data->sr\_no;

return msgSum;

}

int checkSum(struct data\_pkt\* data){

long check = createSum(data);

if(data->checkSum == check)return 1;

return 0;

}

int shakeHand(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,

struct control\_pkt \* c\_pkt){

setZero(c\_pkt);

struct ack\_pkt ackPkt;

struct ack\_pkt \* ack = &ackPkt;

ack->ack=0;

ack->sr\_no=0;

int valid = 0;

while(valid == 0){

if(recvfrom(socket\_desc,c\_pkt,sizeof(\*c\_pkt),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

printf("\nRecved control pkt\n");

valid = checkSumControl(c\_pkt);

ack->ack=1;

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending akc to client.");

return -1;

}

}

printf("\nSent control pkt ack.\n");

return 1;

}

int comunicate(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

ack->ack=1;

printf("\nData pkt is valid\n");

}

else{

ack->ack=0;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

char c;

scanf("%c",&c);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

}

int comunicatePkt(int socket\_desc,struct sockaddr\_in client\_addr,int client\_addr\_size,int sr\_no){

struct data\_pkt data;

struct data\_pkt \* data1 = &data;

memset(data1->msg,0,100);

struct ack\_pkt ackPKT;

struct ack\_pkt \* ack = &ackPKT;

if(recvfrom(socket\_desc,data1,sizeof(\*data1),0,

(struct sockaddr \*)&client\_addr,&client\_addr\_size)<0){

printf("\nRecv error.\n");

return -1;

}

int valid = checkSum(data1);

printf("\nchecksum: %ld",createSum(data1));

printf("\nMsg from client: %s",data1->msg);

if(valid == 1){

if(data1->sr\_no != sr\_no) return -1;

ack->ack=1;

char \* filename = "get.txt";

FILE \*fp = fopen(filename,"a");

fprintf(fp,"%s",data1->msg);

printf("\nmessage: %s",data1->msg);

printf("\nData pkt is valid\n");

}

else{

ack->ack=-1;

printf("\nData pkt is not valid\n");

}

ack->checkSum = createSumACK(ack);

if(sendto(socket\_desc,ack,sizeof(\*ack),0,

(struct sockaddr \*)&client\_addr,client\_addr\_size)<0){

printf("\nError while sending ack to client.");

return -1;

}

return ack->ack;

}

int main(){

int socket\_desc;

struct sockaddr\_in server\_addr,client\_addr;

int client\_addr\_size = sizeof(client\_addr);

socket\_desc = socket(AF\_INET,SOCK\_DGRAM,IPPROTO\_UDP);

if(socket\_desc < 0){

printf("\nError while creating socket.");

return -1;

}

printf("\nCreated socket.");

p

server\_addr.sin\_family = AF\_INET;

server\_addr.sin\_port = htons(PORT);

server\_addr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

if(bind(socket\_desc,(struct sockaddr \*)&server\_addr,sizeof(server\_addr))<0){

printf("\nBinding error");

return -1;

}

printf("\nBinding of socket.");

printf("\nNow listening...");

struct control\_pkt pkt;

int handshakeResult = shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

while(handshakeResult==-1){

handshakeResult=shakeHand(socket\_desc,client\_addr,client\_addr\_size,&pkt);

return -1;

}

printf("\nPackets : %d\n",pkt.no\_of\_pkt);

for(int i=0;i<3;i++){

int result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

while(result == -1){

result = comunicatePkt(socket\_desc,client\_addr,client\_addr\_size,i);

}

}

return 1;

}

Server Screenshot :

