STOP AND WAIT ARQ

CLIENT:-

#include<sys/types.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<netdb.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

int main()

{

int sock,val,i,count,port;

char recvdata[50],sentdata[50];

struct sockaddr\_in server\_addr;

printf("\n\n------Client Running--------");

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("Socket");

exit(1);

}

printf("\nEnter port number:");

scanf("%d",&port);

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

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

server\_addr.sin\_addr.s\_addr=htonl(INADDR\_ANY);

bzero(&(server\_addr.sin\_zero),8);

if(connect(sock,(struct sockaddr \*)&server\_addr,sizeof(struct sockaddr))==-1)

{

perror("Connect");

exit(1);

}

while(1)

{

printf("\n\nEnter frame number : ");

scanf("%d",&val);

send(sock,&val,sizeof(val),0);

printf("Enter data: ");

scanf("%s",sentdata);

send(sock,sentdata,strlen(sentdata),0);

if(strcmp(sentdata,"end")==0)

break;

recv(sock,&count,sizeof(count),0);

i=recv(sock,recvdata,50,0);

recvdata[i]='\0';

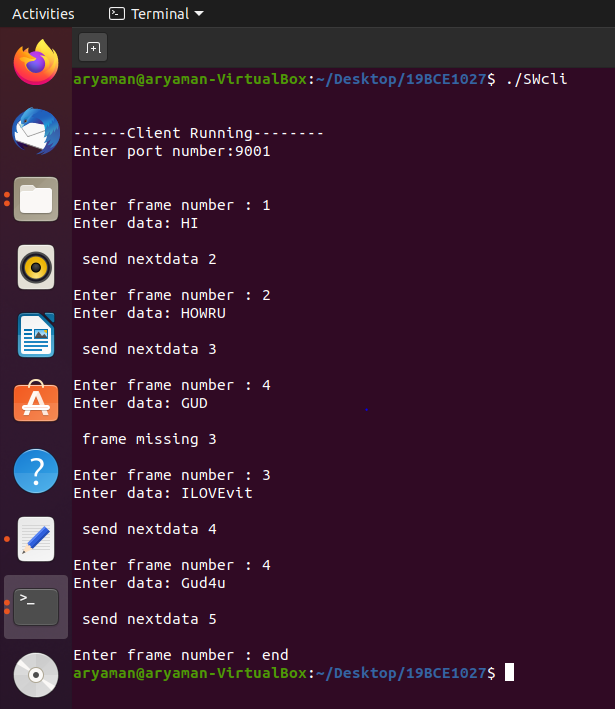
printf("\n %s %d",recvdata,count);

}

close(sock);

return 0;

}



SERVER:-

#include<sys/types.h>

#include<netinet/in.h>

#include<netdb.h>

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<unistd.h>

#include<errno.h>

int main()

{

int sock,size,connect;

char senderdata[50],data[50];

int val,count,i,port;

struct sockaddr\_in ser,cli;

printf("\n\n--------Server Running------");

if((sock=socket(AF\_INET,SOCK\_STREAM,0))==-1)

{

perror("\nSocket Creation Error");

exit(-1);

}

printf("\nEnter the port number:");

scanf("%d",&port);

ser.sin\_family=AF\_INET;

ser.sin\_port=htons(port);

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

bzero(&(ser.sin\_zero),8);

if(bind(sock,(struct sockaddr \*)&ser,sizeof(struct sockaddr))==-1)

{

perror("\n\t Error in Bind");

exit(-1);

}

if(listen(sock,2)==-1)

{

perror("\n\t Error in listen");

exit(-1);

}

printf("\n\t Waiting for Connection");

size=sizeof(struct sockaddr);

connect=accept(sock,(struct sockaddr \*)&cli,&size);

if(connect==-1)

{

perror("\n\t Connection Failed");

exit(-1);

}

printf("\n\t Connected Succesfully");

printf("\n");

recv(connect,&val,sizeof(val),0);

count=val;

while(1)

{

i=recv(connect,&data,sizeof(data),0);

data[i]='\0';

if(strcmp(data,"end")==0)

{

printf("\n\t Finished");

break;

}

if(count!=val)

{

strcpy(senderdata,"frame missing");

send(connect,&count,sizeof(count),0);

send(connect,senderdata,strlen(senderdata),0);

}

else

{

printf("\n The frame number is: %d",val);

printf("\n The data is : %s",data);

count++;

strcpy(senderdata,"send nextdata");

send(connect,&count,sizeof(count),0);

send(connect,senderdata,strlen(senderdata),0);

}

printf("\n The expected frame now is:%d\n",count);

recv(connect,&val,sizeof(val),0);

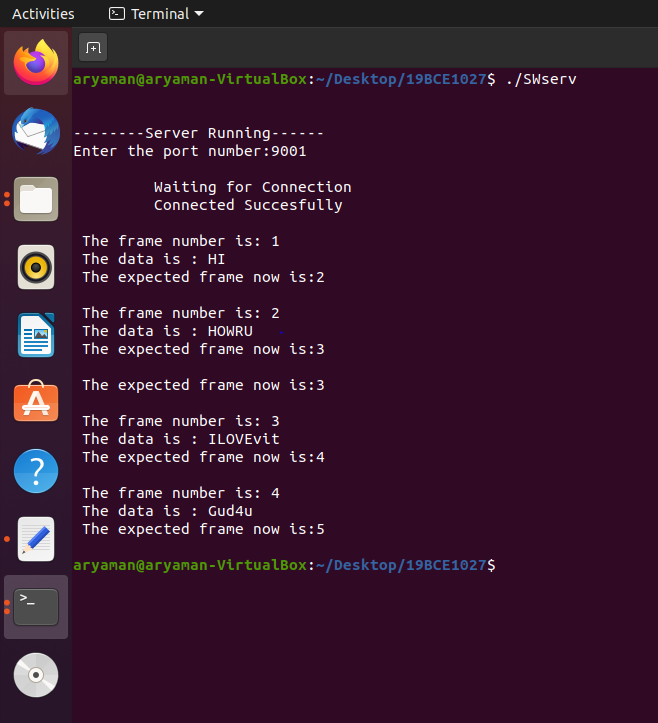
}

close(connect);

close(sock);

return 0;

}



GO-BACK-N-ARQ:-

#include<stdio.h>

int main()

{

int windowsize,sent=0,ack,i;

printf("Enter window size n");

scanf("%d",&windowsize);

while(1)

{

for(i=0;i<windowsize;i++)

{

printf("Frame %d has been transmitted.\n",sent);

sent++;

if(sent==windowsize)

break;

}

printf("\nPlease enter the last acknowledgement received.\n");

scanf("%d",&ack);

if(ack==windowsize)

break;

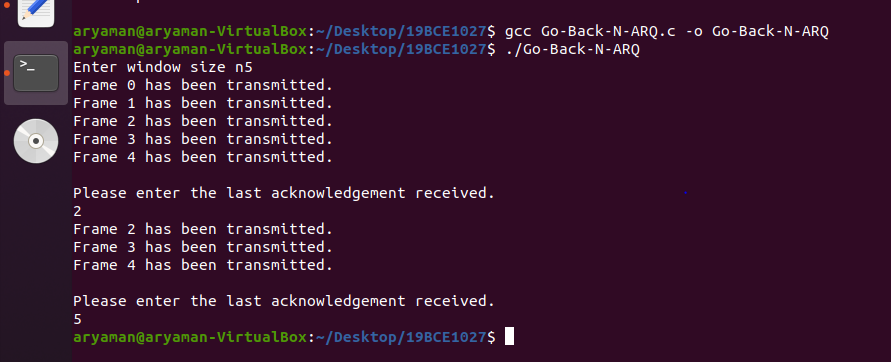
else

sent=ack;

}

return 0;

}



SELECTIVE REPEAT ARQ:

#include<stdio.h>

int main()

{

int w,i,f,frames[50];

printf("Enter window size: ");

scanf("%d",&w);

printf("\nEnter number of frames to transmit: ");

scanf("%d",&f);

printf("\nEnter %d frames: ",f);

for(i=1;i<=f;i++)

scanf("%d",&frames[i]);

printf("\nWith sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)\n\n");

printf("After sending %d frames at each stage sender waits for acknowledgement sent by the receiver\n\n",w);

for(i=1;i<=f;i++)

{

if(i%w==0)

{

printf("%d\n",frames[i]);

printf("Acknowledgement of above frames sent is received by sender\n\n");

}

else

printf("%d ",frames[i]);

}

if(f%w!=0)

printf("\nAcknowledgement of above frames sent is received by sender\n");

return 0;

}

