

```
int sockfd;
struct sockaddr_in serverAddr;
char buffer[1024];
socklen_t addr_size;
int frame_id = 0;
Frame frame_send;
Frame frame_recv;
int ack_recv = 1;
sockfd = socket(AF_INET, SOCK_DGRAM, 0);
memset(&serverAddr, '\0', sizeof(serverAddr));
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(port);
serverAddr.sin_addr.s_addr = inet_addr("127.0.0.1");
while(1){
    if(ack_recv == 1){
        frame_send.sq_no = frame_id;
        frame_send.frame_kind = 1;
        frame_send.ack = 0;
        printf("Enter Data: ");
        scanf("%s", buffer);
        strcpy(frame_send.packet.data, buffer);
        sendto(sockfd, &frame_send, sizeof(Frame), 0, (struct
        sockaddr*)&serverAddr, sizeof(serverAddr));
        printf("[+]Frame Send\n");
    }
    int addr_size = sizeof(serverAddr);
    int f_recv_size = recvfrom(sockfd, &frame_recv, sizeof(frame_recv), 0, (struct
    sockaddr*)&serverAddr, &addr_size);
    if( f_recv_size > 0 && frame_recv.sq_no == 0 && frame_recv.ack ==
    frame_id+1){
        printf("[+]Ack Received\n");
        ack_recv = 1;
    }else{
        printf("[-]Ack Not Received\n");
        ack_recv = 0;
    }
    frame_id++;
}
close(sockfd);
return 0;
```

}

OUTPUT

```

11111 Welcome to Linux Server 11111
Last login: Tue Jun 21 20:04:53 2022 from 192.168.99.215
p1920@administrator-rusa:~$ cd unni/CN/
p1920@administrator-rusa:~/unni/CN$ atom
p1920@administrator-rusa:~/unni/CN$ gcc slide_window.c
p1920@administrator-rusa:~/unni/CN$ ./a.out
Enter window size: 000
Enter number of frames to transmit: 2
Enter 2 frames: 12
qw
With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)
After sending 600 frames at each stage sender waits for acknowledgement sent by the receiver
12 .609933392
Acknowledgement of above frames sent is received by sender
p1920@administrator-rusa:~/unni/CN$ ./a.out
Enter window size: 3
Enter number of frames to transmit: 5
Enter 5 frames: 12 5 89 4 0
With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)
After sending 3 frames at each stage sender waits for acknowledgement sent by the receiver
12 5 89
Acknowledgement of above frames sent is received by sender
4 0
Acknowledgement of above frames sent is received by sender
p1920@administrator-rusa:~/unni/CN$

```

2. Go_Back ARQ

Reciver.c

```

#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];
void alpha9(int);
int main()
{
    struct sockaddr_in ser,cli;
    int s,n,sock,i,j,c=1,f;

```

```
unsigned int s1;
s=socket(AF_INET,SOCK_STREAM,0);
ser.sin_family=AF_INET;
ser.sin_port=6500;
ser.sin_addr.s_addr=inet_addr("127.0.0.1");
bind(s,(struct sockaddr *)&ser, sizeof(ser));
listen(s,1);
n=sizeof(cli);
sock=accept(s,(struct sockaddr *)&cli, &n);
printf("\nTCP Connection Established.\n");
s1=(unsigned int) time(NULL);
srand(s1);
strcpy(b,"Time Out ");
recv(sock,a,sizeof(a),0);
f=atoi(a);
while(1)
{
for(i=0;i<W;i++)
{
recv(sock,a,sizeof(a),0);
if(strcmp(a,b)==0)
{
break;
}
}
i=0;
while(i<W)
{
j=rand()%P1;
if(j<P2)
{
send(sock,b,sizeof(b),0);
break;
}
else
{
alpha9(c);
if(c<=f)
{
printf("\nFrame %s Received ",a);
```

```
send(sock,a,sizeof(a),0);
}
else
{
break;
}
c++;
}
if(c>f)
{
break;
}
i++;
}
}
close(sock);
close(s);
return 0;
}
void alpha9(int z)
{
int k,i=0,j,g;
k=z;
while(k>0)
{
i++;
k=k/10;
}
g=i;
i--;
while(z>0)
{
k=z%10;
a[i]=k+48;
i--;
z=z/10;
}
a[g]='\0';
}
```

Client.c

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<time.h>
#include<stdlib.h>
#include<ctype.h>
#include<arpa/inet.h>
#define W 5
#define P1 50
#define P2 10
char a[10];
char b[10];
void alpha9(int);
int main()
{
    struct sockaddr_in ser,cli;
    int s,n,sock,i,j,c=1,f;
    unsigned int s1;
    s=socket(AF_INET,SOCK_STREAM,0);
    ser.sin_family=AF_INET;
    ser.sin_port=6500;
    ser.sin_addr.s_addr=inet_addr("127.0.0.1");
    bind(s,(struct sockaddr *) &ser, sizeof(ser));
    listen(s,1);
    n=sizeof(cli);
    sock=accept(s,(struct sockaddr *)&cli, &n);
    printf("\nTCP Connection Established.\n");
    s1=(unsigned int) time(NULL);
    srand(s1);
    strcpy(b,"Time Out ");
    recv(sock,a,sizeof(a),0);
    f=atoi(a);
    while(1)
    {
        for(i=0;i<W;i++)
        {
```