

SERVER SIDE

```
//Program for Stop and Wait ARQ_Server
#include<stdio.h>
#include<stdlib.h>
#include<sys/types.h>
#include<netdb.h>
#include<netinet/in.h>
#include<string.h>
#include<unistd.h>
int main()
{
    char msg[50];
    int svrfd,clnfd,err,totfram,fsnd,ack,pno;
    socklen_t sclen;
    struct sockaddr_in server,client;
    printf("\n Enter the port number : ");
    scanf("%d",&pno);
    memset(&server,0,sizeof(server));
    memset(&client,0,sizeof(client));
    svrfd = socket(AF_INET,SOCK_DGRAM,0);
    if(svrfd == -1)
    {
        printf("\n Socket creation error !!! \n");
        exit(0);
    }
    server.sin_family = AF_INET;
    server.sin_port = htons(pno);
    server.sin_addr.s_addr = INADDR_ANY;
    if(bind(svrfd,(struct sockaddr *)&server,sizeof(server)) == -1)
    {
        printf("\n Binding error !!! \n");
        exit(0);
    }
    printf("\n Server waiting for connection ... \n");
    sclen = sizeof(client);
    recvfrom(svrfd,msg,sizeof(msg),0,(struct sockaddr *)&client,&sclen);
    printf("\n Client connected ... \n");
    totfram = 5;
    printf("\n Total packet frame to transmitted \n");
    sendto(svrfd,(char *)&totfram,sizeof(totfram),0,(struct sockaddr *)&client,sclen);
    sleep(1);
    fsnd = 1;
    ack = 1;
    while(fsnd <= totfram)
    {
        if(ack == 1)
            printf("\n Transmitted frame #%d\n",fsnd);
        else
            printf("\n Retransmitting frames #%d\n",fsnd);
        sendto(svrfd,(char *)&fsnd,sizeof(fsnd),0,(struct sockaddr *)&client,sclen);
        printf("\n Waiting for Acknowledgment");
    }
}
```

```

        sleep(1);
        recvfrom(svrfd,&ack,sizeof(ack),0,(struct sockaddr *)&client,&sclen);
        printf("\n Received As \n");
        if(ack == -1)
            printf("\n Negative \n");
        else
        {
            printf("\n Positive \n");
            fsnd++;
        }
        sleep(1);
    }
    printf("\n All frames are transmitted successfully \n Closing Connection \n");
    close(svrfd);
    return 0;
}
//End of the program

```

CLIENT SIDE

```

//Program for Stop and Wait ARQ_Client
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>
int main()
{
    char msg[50];
    int pno,clnfd,totfram = 5,frcvd,bufr,ack,err,first;
    socklen_t sclen;
    struct sockaddr_in client;
    printf("\n Enter the port number : ");
    scanf("%d",&pno);
    memset(&client,0,sizeof(client));
    clnfd = socket(AF_INET,SOCK_DGRAM,0);
    if(clnfd == -1)
    {
        printf("\n Socket creation error !!! \n");
        exit(1);
    }
    client.sin_family = AF_INET;
    client.sin_port = htons(pno);
    client.sin_addr.s_addr = inet_addr("127.0.0.1");
    sclen = sizeof(client);
    printf("\n Sending connection confirmation ... \n");
    sprintf(msg,"ready to receive packet frames ...");
    sendto(clnfd,msg,sizeof(msg),0,(struct sockaddr*)&client,sclen);
}

```

```

printf("\n Waiting for sender response ... \n");
sleep(1);
recvfrom(clnfd,msg,sizeof(totfram),0,(struct sockaddr*)&client,&sclen);
first=1;
bufr=1;
while(bufr <= totfram)
{
    recvfrom(clnfd,&frcvd,sizeof(frcvd),0,(struct sockaddr*)&client,&sclen);
    if(first == 1 && frcvd == 3)
    {
        frcvd++;
        first = 0;
    }
    if(bufr == frcvd)
    {
        ack = 1;
        printf("\n frame # %d receive",frcvd);
        bufr++;
    }
    else
    {
        ack = -1;
        printf("\n frame # %d received",frcvd);
    }
    printf("\n sending ack to server \n");
    if(ack == 1)
        printf("\n Positive");
    else
        printf("\n Negative");
    sendto(clnfd,(char*)&ack,sizeof(ack),0,(struct sockaddr*)&client,sclen);
    sleep(2);
}
printf("\n All frames are transmitted successfully \n Closing connection \n");
close(clnfd);
return 0;
}
//End of the program

```

OUTPUT

SERVER_SIDE

```
user@hp:~/Documents$ gcc sws.c
user@hp:~/Documents$ ./a.out

Enter the port number : 8053
Server waiting for connection ...
Client connected ...
Total packet frame to transmitted
Transmitted frame #1
Waiting for Acknowledgment
Received As
Positive
Transmitted frame #2
Waiting for Acknowledgment
Received As
Positive
Transmitted frame #3
Waiting for Acknowledgment
Received As
Negative
Retransmitting frames #3
Waiting for Acknowledgment
Received As
Positive
Transmitted frame #4
Waiting for Acknowledgment
Received As
Positive
Transmitted frame #5
Waiting for Acknowledgment
Received As
Positive
All frames are transmitted successfully
Closing Connection
user@hp:~/Documents$
```

CLIENT_SIDE

```
user@hp:~/Documents$ gcc swc.c
user@hp:~/Documents$ ./a.out

Enter the port number : 8053

Sending connection confirmation ...

Waiting for sender response ...

frame # 1 receive
sending ack to server

Positive
frame # 2 receive
sending ack to server

Positive
frame # 4 received
sending ack to server

Negative
frame # 3 receive
sending ack to server

Positive
frame # 4 receive
sending ack to server

Positive
frame # 5 receive
sending ack to server

Positive
All frames are transmitted successfully
Closing connection
user@hp:~/Documents$
```

SERVER SIDE

```
//Program for Go Back N ARQ_Server
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<string.h>
```

```
#include<netinet/in.h>
```

```
#include<sys/types.h>
```

```
#include<sys/socket.h>
```

```
#include<netdb.h>
```

```
#include<netdb.h>
```

```
#include<unistd.h>
```

```
#include<fcntl.h>
```

```
void itoa(int number,char number_string[])
```

```
{
```

```
    number_string[0]=(char)(number+48);
```

```
    number_string[1]='\0';
```

```
}
```

```
int main()
```

```
{
```

```
sockfd,pno,newsockfd,size>windowstart=1>windowcurrent=1>windowend=4,oldwindowstart,flag;
```

```
    char buffer[100];
```

```
    socklen_t len;
```

```
    struct sockaddr_in client,server;
```

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

```
    scanf("%d",&pno);
```

```
    memset(&server,0,sizeof(server));
```

```
    memset(&client,0,sizeof(client));
```

```
    if((sockfd=socket(AF_INET,SOCK_STREAM,0))==-1)
```

```
    {
```

```
        printf("\nError in socket creation !!!\n");
```

```
        exit(1);
```

```
    }
```

```
    else
```

```
        printf("\nSocket created ...\n");
```

```
        server.sin_family=AF_INET;
```

```
        server.sin_port=htons(pno);
```

```
        server.sin_addr.s_addr=INADDR_ANY;
```

```
        printf("\nServer started ...\n");
```

```
        if(bind(sockfd,(struct sockaddr *)&server,sizeof(server))==-1)
```

```
        {
```

```
            printf("\nBinding Error !!!\n");
```

```
            exit(1);
```

```
        }
```

```
        else
```

```
            printf("\nBinding Successful ...\n");
```

```
            len=sizeof(client);
```

```
if(listen(sockfd,20)!=-1)
```

```
{
```

```
    if((newsockfd=accept(sockfd,(struct sockaddr *)&client,&len))==-1)
```

int

```

{
    printf("\nError in accepting connection !!!\n");
    exit(1);
}
memset(&buffer,0,sizeof(buffer));
if(recv(newsockfd,buffer,100,0)==-1)
{
    printf("\nReceive error !!!\n");
    exit(1);
}
fcntl(newsockfd,F_SETFL,O_NONBLOCK);
printf("\nReceiving request from client ...\n");
do
{
    if(windowcurrent!=windowend)
    {
        memset(&buffer,0,sizeof(buffer));
        itoa(windowcurrent,buffer);
        send(newsockfd,buffer,100,0);
        printf("\nPacket send:%d ...",windowcurrent);
    }
    windowcurrent++;
    printf("\n%d||%d",windowcurrent,windowend);
    memset(&buffer,'\0',sizeof(buffer));
    if(recv(newsockfd,buffer,100,0)!=-1)
    {
        if(buffer[0]=='R')
        {
            printf("\nReceived a retransmit packet.. resending %c ...",buffer[1]);
            itoa((atoi(&buffer[1])),buffer);
            send(newsockfd,buffer,100,0);
            windowcurrent=atoi(&buffer[0]);
            windowcurrent++;
        }
        else if(buffer[0]=='A')
        {
            oldwindowstart=windowstart;
            windowstart=atoi(&buffer[1])+1;
            windowend+=(windowstart-oldwindowstart);
            printf("\nReceived acknowledgement %c moving window ...",buffer[1]);
        }
    }
    sleep(1);
} while (windowcurrent!=10);}
else
{
    printf("\nError in listening !!!\n");
    exit(1);
}
close(sockfd);
close(newsockfd);
printf("\nSending complete...\n");

```

```
    return 0;
}

//End of the program
```

CLIENT SIDE

```
//Program for Go Back N ARQ _Client
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netinet/in.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netdb.h>
#include<netdb.h>
#include<unistd.h>
int main()
{
    char data[100],digit[2];
    int sockfd,new_sockfd,size,first_time=1,current_packet,wait=3,pno;
    struct sockaddr_in client;
    memset(&client,0,sizeof(client));
    printf("\nEnter the port number : ");
    scanf("%d",&pno);
    sockfd=socket(AF_INET,SOCK_STREAM,0);
    if(sockfd==-1)
    {
        printf("\nError in socket creation !!!\n");
        exit(1);
    }
    else
        printf("\nSocket created ... \n ");
    client.sin_family=AF_INET;
    client.sin_port=htons(pno);
    client.sin_addr.s_addr=INADDR_ANY;
    printf("\nClient started ... \n");
    size=sizeof(client);
    if(connect(sockfd,(struct sockaddr *)&client,size)==-1)
    {
        printf("\nError in connecting to server !!! \n");
        exit(1);
    }
    else
        printf("\nConnection establised !!! \n");
    memset(&data,0,sizeof(data));
    sprintf(data,"REQUEST");
    if(send(sockfd,data,strlen(data),0)==-1)
    {
        printf("\nError in sending !!! \n");
        exit(1);
    }
}
```



```

do
{
    memset(&data,0,sizeof(data));
    recv(sockfd,data,100,0);
    current_packet=atoi(data);
    printf("\nGot packet : %d ",current_packet);
    if(current_packet==3 && first_time)
    {
        printf("\nSimulation : packet data corrupted or incomplete !!!\nSending
retransmit for packet %d...\n",current_packet);
        memset(&data,0,sizeof(data));
        sprintf(data,"R");
        if(send(sockfd,data,strlen(data),0)==-1)
        {
            printf("\nError in sending !!! \n");
            exit(1);
        }
        first_time=0;
        wait--;
        if(!wait)
        {
            printf("\nPacket accepted ---> sending acknowledgement ...\n");
            memset(&data,0,sizeof(data));
            digit[0]=(char)(current_packet+48);
            digit[1]='\0';
            strcat(data,digit);
            send(sockfd,data,strlen(data),0);
        }
    }

}while(current_packet!=9);
printf("\nAll packets received ... \nExiting ...\n");
close(sockfd);
return 0;
}
//End of the program

```

OUTPUT

SERVER_SIDE

```
File Edit View Search Terminal Help
user@hp:~/Documents$ gcc gbn_server.c
user@hp:~/Documents$ ./a.out

Enter the port number : 3033

Socket created ...

Server started ...

Binding Successful ...

Receiving request from client ...

Packet send:1 ...
2||4
Packet send:2 ...
3||4
Packet send:3 ...
4||4
5||4
Received a retransmit packet.. resending ...
Packet send:1 ...
2||4
Packet send:2 ...
3||4
Packet send:3 ...
4||4
5||4
Packet send:5 ...
6||4
Packet send:6 ...
7||4
Packet send:7 ...
8||4
Packet send:8 ...
9||4
Packet send:9 ...
10||4
Sending complete...
user@hp:~/Documents$
```

CLIENT_SIDE

```
user@hp: ~/Documents
File Edit View Search Terminal Help
user@hp:~/Documents$ gcc gbn_client.c
user@hp:~/Documents$ ./a.out

Enter the port number : 3033

Socket created ...

Client started ...

Connection established !!!

Got packet : 1
Got packet : 2
Got packet : 3
Simulation : packet data corrupted or incomplete !!!
Sending retransmit for packet 3...

Got packet : 0
Got packet : 1
Got packet : 2
Got packet : 3
Got packet : 5
Got packet : 6
Got packet : 7
Got packet : 8
Got packet : 9
All packets received ...
Exiting ...
user@hp:~/Documents$
```

SERVER SIDE

```
//Program for Selective Repeat ARQ_Server
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netdb.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<sys/socket.h>
#include<fcntl.h>
#include<unistd.h>

void itoa(int num,char nums[])
{
    nums[0]=(char)(num+48);
    nums[1]='\0';
}

int main()
{
    int pno,sfd,newsfd,size,ws=1,wc=1,we=4,o,flag;
    char buf[100];
    socklen_t len;
    struct sockaddr_in server,client;
    memset(&server,0,sizeof(server));
    memset(&client,0,sizeof(client));
    printf("\n Enter the port number : ");
    scanf("%d",&pno);
    if((sfd=socket(AF_INET,SOCK_STREAM,0))==-1)
    {
        printf("\nError in socket creation !!!\n");
        exit(1);
    }
    else
        printf("\n Socket created ...\n");
    server.sin_family=AF_INET;
    server.sin_port=htons(pno);
    server.sin_addr.s_addr=INADDR_ANY;
    printf("\n Server Started ... \n");
    if(bind(sfd,(struct sockaddr *)&server,sizeof(server))==-1)
    {
        printf("\n Binding Error !!!\n");
        exit (1);
    }
    else
        printf("\n Binding successful ... waiting for connection ...\n");
    len=sizeof(client);
    if(listen(sfd,20)!=-1)
    {
        if((newsfd=accept(sfd,(struct sockaddr *)&client,&len))==-1)
        {
            printf("\nError in accepting connection !!!\n");
```

```

        exit(1);
    }

    memset(&buf,0,sizeof(buf));

    if(recv(newsf, buf, 100, 0) == -1)
    {
        printf("\n Receive error !!!\n");
        exit(1);
    }
    fcntl(newsf, F_SETFL, O_NONBLOCK);
    printf("\n Received request from client ... sending packet ...\n");
    do
    {
        if(wc != we)
        {
            memset(&buf, 0, sizeof(buf));
            itoa(wc, buf);
            send(newsf, buf, 100, 0);
            printf("\n Packet send: %d ...", wc);
            wc++;
        }
        printf("\n **%d||%d**", wc, we);
        memset(&buf, '\0', sizeof(buf));
        if(recv(newsf, buf, 100, 0) != -1)
        {
            if(buf[0] == 'R')
            {
                printf("\n Received a retransmit packet ... resending packets
no: %c ...", buf[1]);

                itoa((atoi(&buf[1])), buf);
                send(newsf, buf, 100, 0);
                o = ws;
                ws = atoi(&buf[1]) + 1;
                we += (ws - o);
                printf("\n **Received ACK %c ...moving window
boundary ...", buf[1]);

                wc++;
            }
            else
            if(buf[0] == 'A')
            {
                o = ws;
                ws = atoi(&buf[1]) + 1;
                we = we + (ws - o);
                printf("\n Received ACK %c.. moving window
boundary ...", buf[1]);

                wc++;
            }
        }
        sleep(1);
    } while(wc != 10);

```

```

    } else {
        printf("\n Error in listening !!!\n");
        exit(1);
    }
    close(sfd);
    close(newsfdf);
    printf("\n Sending complete ... socket closed exiting...\n");
    return 0;
}
//End of the program

```

CLIENT SIDE

```

//Program for Selective Repeat ARQ_Client
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netdb.h>
#include<sys/types.h>
#include <arpa/inet.h>
#include<netinet/in.h>
#include<sys/socket.h>
#include<unistd.h>
int main()
{
    int pno,sfd,newsfdf,size,ft=1,curr,wait=3;
    char data[100],digit[2];
    socklen_t len;
    struct sockaddr_in client;
    printf("\n Enter the port number : ");
    scanf("%d",&pno);
    memset(&client,0,sizeof(client));
    if((sfd=socket(AF_INET,SOCK_STREAM,0))==-1)
    {
        printf("\n Error in socket creation !!!");
        exit(1);
    }
    else
        printf("\n Socket created ...\n");
    client.sin_family=AF_INET;
    client.sin_port=htons(pno);
    client.sin_addr.s_addr=inet_addr("127.0.0.1");
    printf("\n Client started ...");
    size=sizeof(client);
    printf("\n Establishing connection ...\n");
    if(connect(sfd,(struct sockaddr*)&client,size)==-1)
    {
        printf("\n Error in connecting to server !!! \n");
        exit(1);
    }
    else

```

```

        printf("\n Connection Established ... \n");
memset(&data,0,sizeof(data));
sprintf(data,"REQUEST");
if(send(sfd,data,strlen(data),0)==-1)
{
    printf("\n Error in sending request for data !!!");
    exit(1);
}
do
{
    memset(&data,0,sizeof(data));
    recv(sfd,data,100,0);
    curr=atoi(data);
    printf("\n Got packet : %d...",curr);
    if(curr==3 &&ft)
    {
        printf("\n Simulation:packet data corrupted or incomplete !!!\n Sending
retransmit for the packet ...");
        memset(&data,0,sizeof(data));
        sprintf(data,"R3");
        if(send(sfd,data,strlen(data),0)==-1)
        {
            printf("\n Error in sending retransmit !!!");
            exit(1);
        }
        ft=0;
    }
    else
    {
        wait--;
        if(!wait)
        {
            printf("\n Packet accepted -> sending ACK ...");
            wait=3;
            memset(&data,0,sizeof(data));
            sprintf(data,"A");
            digit[0]=(char)(curr+48);
            digit[1]='\0';
            strcat(data,digit);
            send(sfd,data,strlen(data),0);
        }
    }
} while(curr!=8);
printf("\n\n All packets received ...exiting ...\n");
close(sfd);
return 0;
}
//End of the program

```

OUTPUT

```
user@hp: ~/Documents
File Edit View Search Terminal Help
user@hp:~/Documents$ gcc sr_server.c
user@hp:~/Documents$ ./a.out

Enter the port number : 1553

Socket created ...

Server Started ...

Binding successful ... waiting for connection ...

Received request from client ... sending packet ...

Packet send:1 ... add new add along the ... wait ...
**2||4**
Packet send:2 ... ...
**3||4**
Packet send:3 ...
**4||4**
Received a retransmit packet ... resending packets no:3 ...
**Received ACK ...moving window boundary ...

Packet send:5 ...
**6||4**
Received a retransmit packet ... resending packets no:3 ...
**Received ACK ...moving window boundary ...

Packet send:7 ... in_addr_s_addr=inet_addr("127.0.0.1");
**8||4**
Packet send:8 ...
**9||4**
Received a retransmit packet ... resending packets no:7 ...
**Received ACK ...moving window boundary ...

Sending complete ... socket closed exiting...
```

```
user@hp: ~/Documents
File Edit View Search Terminal Help
user@hp:~/Documents$ gcc sr_client.c
user@hp:~/Documents$ ./a.out

Enter the port number : 1553

Socket created ...

Client started ...
Establishing connection ...

Connection Established ...

Got packet : 1...
Got packet : 2...
Got packet : 3...
Simulation:packet data corrupted or incomplete !!!
Sending retransmit for the packet ...
Got packet : 3...
Packet accepted -> sending ACK ...
Got packet : 5...
Got packet : 3...
Got packet : 7...
Packet accepted -> sending ACK ...
Got packet : 8...

All packets received ...exiting ...
user@hp:~/Documents$
```