Exp No: 7 : Sliding window protocol to implement flow control.

Date: 1.9.2025

Aim: write a program to implement flow control at data link layer using sliding window protocol. Simulate the flow of frames from one node to another.

Create a sender program with the following features:

1. Input window size from the user.
2. Input a text message from the user.
3. Consider 1 character per frame.
4. Create a frame with frame no and data.
5. Send the frames.
6. wait for the acknowledgement from the receiver.
7. Reads a file called receive buffer.
8. check Ack field for the acknowledgment number.

7. If the acknowledgment number is expected and new set of frames accordingly, else if NACK is received.

Create a receiver file with the following features.

1. Reades a file called Sender - Buffer
2. Check the frame no.

3. If the frame no is expected, write the appropriate Ack no in the receiver buffer file else write wack. no in the receiver buffer file.

sliding window protocol for sender.

```
#define SF "Senders buffer txt";
void send_frames(int start, int end, char msg)
{
  file *f = fopen(SF, "w");
  for(int i=start; i<end; i++)
  printf(f, "%d %c\n", i, msg[i]);
  fclose(f);
}
int main(){
    int win, base=0;
  char msg[10,0];
  printf("window size:");
  scanf("%d", &win);
  getchar();
```

```
int n=strlen(msg));
while(base<n){
int end=base+win;
send_frames(base, end, msg);
printf("sent frames %d to %d\n",
                    base, end-1);
step(2);
char type[10];
if(!get_ack(type,win)) continue;
  else if(!strcmp(type,"NACK"))
                    base=min;
}
printf("All frames sent\n");
}
```

Sliding window protocol for receiver:-

```
#define RF "receiver_buffer.txt"
void send_ack(const char *type, int n){
    FILE *f=fopen(RF,"w");
    fprintf(f,"%s %d",type,n);
    fclose(f);
}
int main(){
int esp=0;
while(1){
    int fin[100]; char id[100];
    int c=read_frame(fin,id);
    if(!c){
```

```
        Sleep(1)
        Continue;
    }
for(int i=0; i<c; i++){
    if(fr[i] == exp){
    printf("Got frame %d: %c\n", fr[i],
        exp++; Send_ack("ACK", exp);
    }
    else{
    printf("Wrong frame %d,
                expected %d\n";
    Send_ack("NACK" exp);
        break;
    }
}
Sleep(5)
}
}
```

Sample Input:-

Enter the no of frames to send:-7

Enter window size:3

Output:-

sending frames: 012

Receiver: frame 0 received → sending ack 0

Receiver: frame 1 received → sending ack 1

receiver - frame 2 received → sending ack 2

Result:-

Hence the sliding window protocol is completed and executed successfully.