Sliding window Protocol

AIM:

wap to implement flow control at data link layer using SWP. Simulate the flow of frames from one node to another.

Sender program

It reads the window size & message from the user, Constructs frame with seq no:s & data, & writes them to Sender_buf.txt. It then waits for a response in receiver_buffer.txt.

Receiver program

Receiver reads from Sender_buffer.txt, checks if seq no:s are in order & writes the next expected seq no:s / NACK into Receiver_buffer.txt. Errors can be simulated by manually modifing frame numbers.

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
void sender() {
    int win_size, i=0, base=0, n;
    char msg[50];
    File *fp;
    printf("enter window size:");
    scanf("%d", &win_size);
    printf("enter msg:");
```

```
while (base < n) {
    fp = fopen ("send_buf.txt", "w");
    for (i=0; i< winsize && (base+i) < n; i++) {
        fprintf (fp, "%d %d\n", (base + i) % 2,
                 message [base+i]);

    }
    fclose (fp);

    int ack;
    fscanf (fp, "%d", &ack);
    fclose (fp);
    if (ack == (base % 2)) {
        base += winsize;
    } else {
        printf ("NACK received, resending...\n");
    }
}
}
void receiver () {
    FILE *sfp, *rfp;
    char data;
    int exp_inj = 0;
    while (1) {
        sfp = fopen ("send_buf.txt", "r");
        if (sfp == NULL) {
            Sleep(1);
            continue;
        }
        rfp = fopen ("receiver_buf.txt", "w");

    }
    fprintf (rfp, "%d\n", expected % 2);
    fclose (rfp);
    fclose (sfp);
```

```
int main () {
    int
    printf
    &scan
    if
    else
    retu
}

Input
Window si
Message :
Output
Send_buf

[
    {"fra
    {"fra
    {"fr
]
Receive
[
    {
    {
    {
]
```

```
        sleep(1);
    }
}

int main() {
    int role;
    printf("enter 1 for sender, 2 for receiver:");
    scanf("%d", &role);
    if (role==1) sender();
    else if (role==2) receiver();
    return 0;
}
```

Input

Window size : 3
Message : Hello

Output

Send_buffer (after 1st transmission)

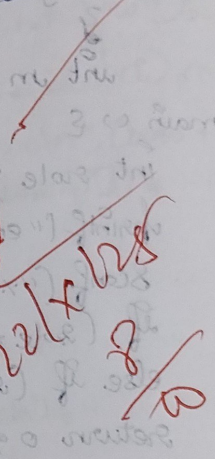
```
[
  {"frame_no": 0, 'data': 'H'},
  {"frame_no": 1, 'data': 'e'},
  {"frame_no": 2, "data": "l"}
]
```

Receiver Buffer (after error detection)

```
[
  {"frame_no": 0, "ack": "ACK"},
  {"frame_no": 1, "ack": "ACK"},
  {"frame_no": 2, "ack": "NACK"},
]
```

Result:

Hence, the code is executed and implemented Successfully.



Exp No: 08
[A]

AIM:

To

on the T

tryHackMe.

Introductio

When tas

tool to

on is

live com

This roo

dedicated

running

rooms

This

Nmap

of the

- Nmap
- Nmap
- Nmap
- Nmap

We

nmas-sc

part

Now,