

27/9/21

## Practical - 7

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 following features.

⇒ Input window size from the user.

⇒ Input a text message from the user.

⇒ Consider 1 character per frame.

⇒ Create a frame with following fields

⇒ Send the frames (print the output on screen and call sender buffer)

⇒ Read a file called Receiver Buffer.

⇒ Check Ack field for the Acknowledgment

Create a receiver file

1. Read a file called Receiver-Buffer

2. Check the frame no.

3. if the frame number is expected write the appropriate Ack no. in the Receiver-Buffer file.

Else write Next no. in the Receiver

(Call function to read the file)

(Call function to write the file)

(Call function to check the frame number)

(Call function to check the Ack field)

def sender():

windowSize = int(input("Enter the window size"))

message = input("message:")

frames = [{"Frame No.": i, "Data": char} for i, char in enumerate(message)]

sender-buffer = []

current-frame = 0

while current-frame < total-frame:

for i in range(window-size):

if current-frame + i < total-frame:

frame = frames[current-frame + i]

sender-buffer.append(frame)

Print(f"Sent : frame No {frame

'Frame No.']) & data: {frame "data"]

with open("sender-buffer.txt", "w") as f:

for frame in sender-buffer:

f.write(f"Frame No. {frame No.}"]

Data: {frame["Data"]}\n")

time.sleep(2)

with open("Receiver-Buffer.txt", "r") as

receiver-buffer = f.read().lines()

```

ack_numbers = []
for line in receiver_response:
    ack_info = line.strip().split()
    if ack_info[0] == "Ack":
        ack_numbers.append(int(ack_info[1]))
    elif ack_info[0] == "NACK":
        nack = int(ack_info[1])
        current_frame = nack_numbers
        break
if set(ack_numbers) == set(range(
    current_frame, current_frame +
    window_size

```

receiver.py

```

from import random
import time

```

```

def receiver():
    while True:
        with open("Sender_Buffer.txt", "r") as file:
            frames = file.readlines()
            receiver_buffer = []
            for frame in frames:
                if random.random() < 0.2:
                    print(f"Frame {frame}")
                    receiver_buffer.append(f"ACK {frame}")
                    break

```

else:

```
Print(f"Received FrameNo. {frame_no} with data: {data}")
receiver_buf.append(f"Ack {frame_no}")
```

```
with open("Receiver_Buffer.txt", "w") as f:
```

```
    # for response in receiver_buf:
    file.write(f"{response}\n")
```

```
time.sleep(5)
```

```
if __name__ == "__main__":
```

```
    receiver()
```

Output

message: hi

Sent: FrameNo. 0 Data: h

Sent: FrameNo. 0 Data: i

receiver

output

Ack 0

Ack: 1

NACK 0

Result

Thus the SD using window Program executed  
Successfully

6/9/24