Ex: NO.7

Date: 12/9/24

practical - 7

Aim: Program should achieve at least below given requirements, you can make it a bidirectional program where is it a bidirectional program where is reciever is sending its data frames with acknowledgement.

Program: sender.py

```
import OS
def sender (window_size, message):
        Sender_buffer = "Sender_Buffer.txt"
        receiver_buffer = "Receiver_Buffer.txt"
        frame_no = 0
    frames = [[i, message[i]] for i in
range (len (message))]
        while frame_no < len (frames):
            for i in range (window_size):
                if frame_no + i < len (frames):
                    print (f'sending frame:
    { frames [frame_no + i }}" }
with open (Sender_buffer, 'a') as f:
    f.write (f'{frames [frame_no + i] [0]}
    {frames [frame_no+i] [1] \n")
   time.sleep (1)
while True:
    if OS.path.exists (receiver_buffer):
        with open (receiver_buffer, 'r') as f:
            ack_no = int (f.read ().strip ())
        os.remove (receiver_buffer)
        break.
    if ack_no >= frame_no:
    print (f"Ack received for frame: {ack
    frame_no = ack_no + 1
```

```
else;
      print(f"NACK received for frame: {frame-no};
   resending ... ")

if_name_ == "_main_"
      window_size = int (input("Enter window
                                     size"))
      message = input("Enter message")
      Sender (window_size, message)

Receiver.py
import os
def receiver():
      Sender_buffer = "Sender_Buffer.txt"
      receiver_buffer = "Receiver_Buffer.txt"
      expected_frame_no=0.
while true:
      if os.path.exist (Sender_buffer):
            with.open (sender_buffer, 'r') as f:
                  lines = f.readlines ()
            os.remove (sender_buffer)

    for line in lines:
            frame = line.strip().split()
            frame_no = int (frame[0])
            data = frame [1]

  if frame_no == expected_frame_no:
            print (f"Received frame: {frame_no
   data: {data}")
      with open (receiver_buffer, 'w') as f:
            f.write (str(frame_no))
            expected_frame_no += 1
      else;
```

```
print (f'unexpected frame : {frame_no},
 data : {data}")
 with open (reciver_buffer, 'W') as f:
    f.write (str (expected_frame_no-1))
if _name_ == "_main_".
    receiver()
```

output :   python Sender.py

```
Enter window size : 5

Enter message : hello

Sending frame : [0, 'h']

Sending frame : [1, 'e']

Sending frame : [2, 'l']

Sending frame : [3, 'l']

Sending frame : [4, 'o']

NACK received for frame :0, resending

Sending frame : [0, 'h']

Sending frame : [4, 'e']

Sending frame : [2, 'l']
Sending frame : [3, '1']
Sending frame : [4, 'o']
NACK received for frame : 0; resending...
Sending frame : [0, 'H']
Sending frame : [1, 'e']
Sending frame : [2, 'l']
Sending frame : [3, 'l']
Sending frame : [4, 'o']

ACK received for frame : 4

python reciever.py

unexpected frame : 2, expected : 0
unexpected frame : 2, expected : 0
unexpected frame : 3, expected : 0
```

Result :

Thus, the program was successfully executed & the output is verified.