THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

EXP 4

CN-C32-2103164

Aim :- Implementation of Go back n sliding window.

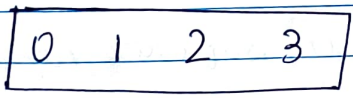
Theory :-

- Discard all subsequent frames following the damaged frames sending no ACKs.
- Eventually the sender times out and retransmits all the unacknowledged frames in order starting with the damaged or lost one.

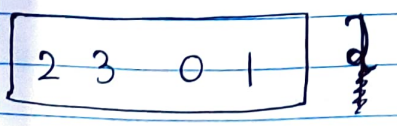
Operation :-

Assume no of bits in sequence no 2.
Window size :- 4 frames.

- Sending window



- Sender has sent 4 frames 0 to 3 and waits for an ACK.
- ACK 0 and ACK 1 received by the sender.



- Now sent 0 1

- Frame 2 lost not received but received 3
- Reciver discards 3 0 and 1

Note

A receiving station does not acknowledge each received frame explicitly. If a sending station received an ack for frame j' and recieves an ack for frame K it assumes frames between j and K have been received correctly

→ Advantages :- Reduces no of acks and lessens n/w traffic.

→ What should be max sender's and recivers window size?

→ Assume K → no of bits in sequence no.

→ Frames are numbered from 0 to $2^K - 1$.

→ Window size cannot be larger than 2^K.

Case 1: Window size larger than 2^K CN-C32-2103164

- Let K = 3
- Assume window size = 9
- Sender window → [0 1 2 3 4 5 6 7 0] 1 2 3 ...

- Problem →
Sender receives an ACK 0. Does not know if it was for first / last frame.

Window size must be less than or equal to 2^K

Case 2: Window size = 2^K

) At time t1 A sends frames 0–7 to B
) B receives each one recently
) At time t2, B sends ACK for the most recived frame ACK 7
) ACK get lost
) Next frame expected by B is frame numbered 0
) It does not receive ACK and hence retransmits frames 0 through 7 at time t3.
) At t4 B recieves frame 0. Sequence no matches with the one it is expecting. Hence B accepts it

Protocal fails (since B has accepted a dulicipate and not a new frame)

Reason

- 2 consective windows contain the same sequence no.
- Solution reduce the sender's window size

Case 3 :- Window size less than 2^K.

Sender's window → [0 1 2 3 4 5 6] 0 1 2

1.) A sends frame 0. through 6 at t 1.
2.) B recives each one correctly.
3.) At time t2, B sends ACK for the most recent recieved frame ACK 6.
4.) ACK gets lost
5.) Nesct frame expected by B is frame num

6.) A does not recive ACK and hence retra frame 0 through 6 at time t3

7.) B recives frame 0 through 6 at t4.
8.) B excepts frame 7.
9.) Hence ignores them.

10.) Eventually B and another ACK 6 whi A recives and A advances it window t include 7. 0 1 2 3 4 5 and protoco continuous.

Operation performed

- Reciever sends a positive ACK if frame has arrived without error and in order (with excepted seq number)
- Reciever does not have to acknowledge each individual frame recieved correctly and in order.
- Reciever can send cumulative ACK for several frames (ACK 4 acknowledges frames (0, 1, 2, 3, 4))
- If frame is damaged or out of order, the reciver discards it and also discards all subsequent frames until it recieves the one excpted.
- In this case, no ACK will be transmitted.
- If the sender timer expires before recieving ACK, it will resend ALL frames beginning with one expired until last one set.

Drawback Gobackn :-

- Reciver discard all correct frames transmitted after bad one.
- Channel bandwidth wasted on retransmitted frames

Alternative Stragery :-

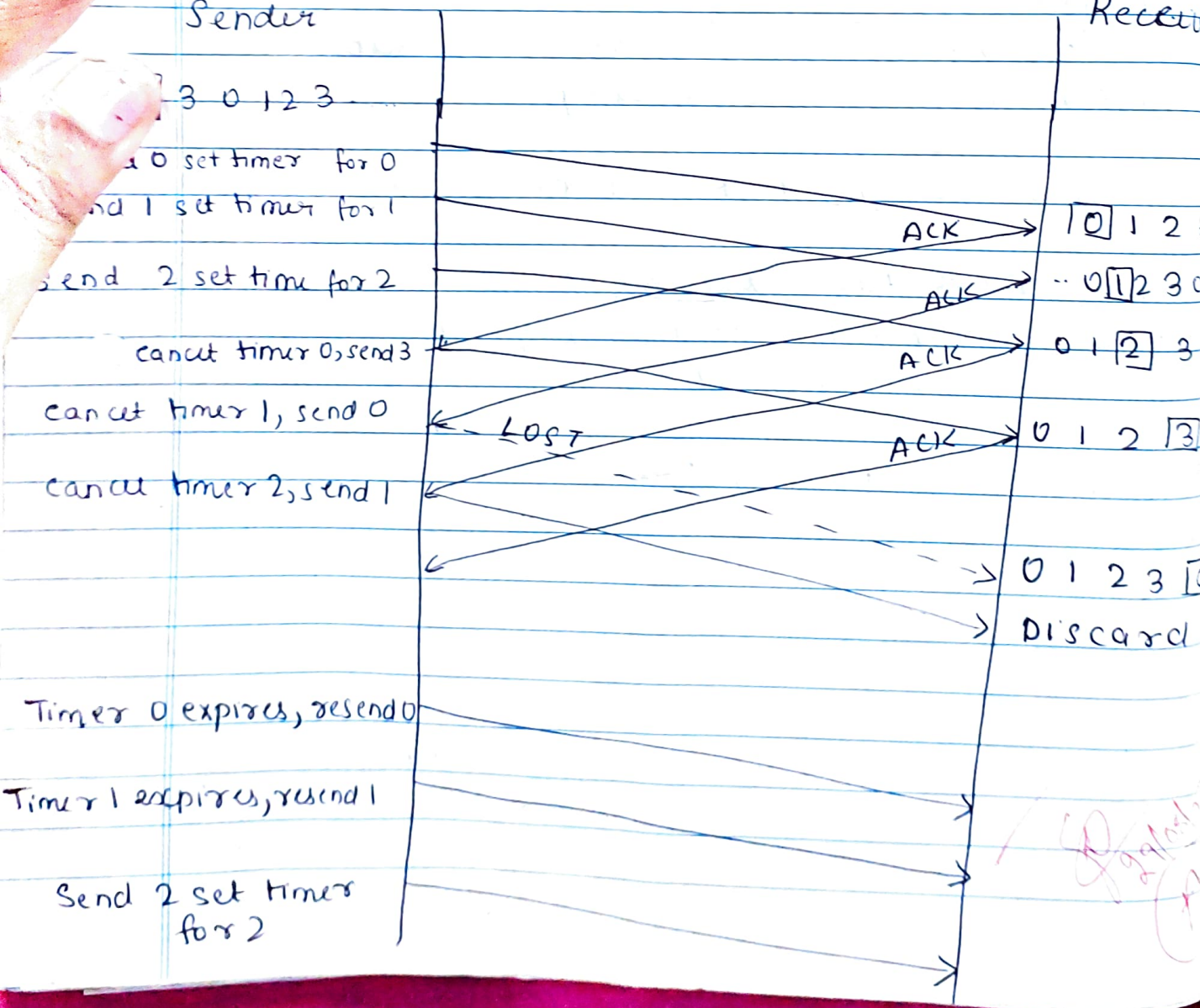
Allow the reciver to accept and buffer the frams following a damaged / lost one - Called Selective Repeat.

Summary

→ Sender window's size - strictly less than 2
(1 less than MAX SEQ)

→ What should be the receivers window s
frames are always received in order.

Hence Receiver's window size should not b
greater than 1.



```
def main():
    print("Enter window size: ")
    window_size = int(input())
    print("Enter total frames to be sent: ")
    total_frames = int(input())

    # Initializing array with data frames
    sender_frames = [i for i in range(total_frames)]

    # Displaying data frames
    for frame in sender_frames:
        print(frame, "| ", end="")
    print()

    # Displaying sender window
    print("Do you want to start sending frames (0/1): ")
    choice = int(input())
    print()

    if choice == 1:
        ptr_on_window_left_sender = 0
        ptr_on_window_left_receiver = 0
        total_sent_frames = 0

        while ptr_on_window_left_sender < total_frames:
            # Sender side
            count = 0
            print("At Sender End:")
            for i in range(ptr_on_window_left_sender, total_frames):
                if count < window_size:
                    print("Sent frame[", i + 1, "]")
                    ptr_on_window_left_sender += 1
                    total_sent_frames += 1
                    count += 1
```

```
            else:
                break

        print()

        # Receiver side
        print("At Receiver end:")
        j = 0
        count = 0
        for i in range(ptr_on_window_left_receiver, total_frames):
            if count < window_size:
                print("Did you receive frame[", i + 1, "] (y/n) : ")
                y_n = input()
                if y_n == 'n':
                    print("Frames will again be sent from frame no.", i + 1)
                    print()
                    ptr_on_window_left_sender = i
                    break
                else:
                    j += 1
                    ptr_on_window_left_receiver += 1
                count += 1
            else:
                break

        if j == window_size:
            print("All Frames from this window sent without errors. Sending next frames...")
            print()

    print()
    print("All frames are sent.")
    print("Total no. of frames sent including retransmission is", total_sent_frames)

if __name__ == "__main__":
```

main()

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\CN> python -u "d:\CN\GoBackN.py"
Enter window size:
4
Enter total frames to be sent:
8
0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
Do you want to start sending frames (0/1):
1

At Sender End:
Sent frame[ 1 ]
Sent frame[ 2 ]
Sent frame[ 3 ]
Sent frame[ 4 ]

At Receiver end:
Did you receive frame[ 1 ] (y/n) :
y
Did you receive frame[ 2 ] (y/n) :
y
Did you receive frame[ 3 ] (y/n) :
y
Did you receive frame[ 4 ] (y/n) :
y
All Frames from this window sent without errors. Sending next frames...
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS


At Sender End:
Sent frame[ 5 ]
Sent frame[ 6 ]
Sent frame[ 7 ]
Sent frame[ 8 ]

At Receiver end:
Did you receive frame[ 5 ] (y/n) :
y
Did you receive frame[ 6 ] (y/n) :
y
Did you receive frame[ 7 ] (y/n) :
n
Frames will again be sent from frame no. 7

At Sender End:
Sent frame[ 7 ]
Sent frame[ 8 ]

At Receiver end:
Did you receive frame[ 7 ] (y/n) :
y
Did you receive frame[ 8 ] (y/n) :
y

All frames are sent.
Total no. of frames sent including retransmission is 10
PS D:\CN>
```