**Experiment 3:**

# Aim:

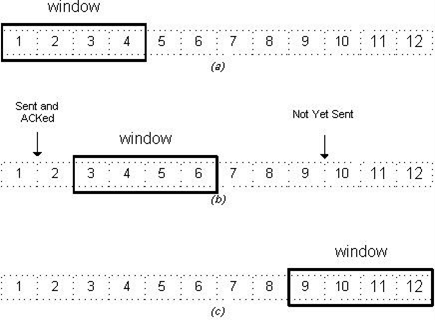
# To write a program developing a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

### **Sliding Window Protocol** : It is a method to transmit data on a network and is applied on the Data Link Layer of OSI model. Window simply means a buffer which has data frames that needs to be transmitted.

Both sender and receiver agrees on some window size. If window size=w then after sending w frames sender waits for the acknowledgement (ack) of the first frame.

As soon as sender receives the acknowledgement of a frame it is replaced by the next frames to be transmitted by the sender. If receiver sends a collective or cumulative acknowledgement to sender then it understands that more than one frames are properly received.

eg:- if ack of frame 3 is received it understands that frame 1 and frame 2 are received properly.



The receiver has to have some memory to compensate any loss in transmission or if the frames are received unordered

Sliding window works in full duplex mode It is of two types:-

1. **Selective Repeat:**Sender transmits only that frame which is erroneous or is lost.
2. **Go back n:**Sender transmits all frames present in the window that occurs after the error bit including error bit
3. **Sliding Window Protocol**

#include <stdio.h>

int main()

{

int w, i, f, frames[50];

printf("Enter window size: ");

scanf("%d", &w);

printf("Enter number of frames to transmit: ");

scanf("%d", &f);

printf("Enter %d frames: ", f);

for (i = 0; i < f; i++)

scanf("%d", &frames[i]);

printf("\nWith sliding window protocol, the frames will be sent in the following manner (assuming no corruption of frames)\n\n");

printf("After sending %d frames at each stage, sender waits for acknowledgement sent by the receiver\n\n", w);

for (i = 0; i < f; i++) {

printf("%d ", frames[i]);

if ((i + 1) % w == 0) {

printf("\nAcknowledgement of above frames sent is received by sender\n\n");

}

}

// If remaining frames are less than window size

if (f % w != 0)

printf("\nAcknowledgement of above frames sent is received by sender\n");

return 0;

}

##### **Output:**

Enter window size: 3

Enter number of frames to transmit: 5

Enter 5 frames: 12 5 89 4 6

With sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)

After sending 3 frames at each stage sender waits for acknowledgement sent by the receiver 12 5 89

Acknowledgement of above frames sent is received by sender 4 6

Acknowledgement of above frames sent is received by sender

**(b)Loss recovery using Go-Back- N Mechanism:**

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <vector>

#include <iomanip>

using namespace std;

const int TOTAL\_PACKETS = 10;

const int WINDOW\_SIZE = 4;

const float LOSS\_PROBABILITY = 0.2f; // 20% chance a packet is lost

bool isPacketLost() {

return (static\_cast<float>(rand()) / RAND\_MAX) < LOSS\_PROBABILITY;

}

int main() {

srand(static\_cast<unsigned int>(time(0)));

int base = 0;

int nextSeqNum = 0;

cout << "Simulating Go-Back-N ARQ Protocol\n";

cout << "Total Packets : " << TOTAL\_PACKETS << "\n";

cout << "Window Size : " << WINDOW\_SIZE << "\n";

cout << "Loss Probability : " << LOSS\_PROBABILITY \* 100 << "%\n\n";

while (base < TOTAL\_PACKETS) {

cout << "Sending window: [";

for (int i = 0; i < WINDOW\_SIZE && (base + i) < TOTAL\_PACKETS; ++i) {

cout << " " << base + i;

}

cout << " ]\n";

bool lossOccurred = false;

int i;

for (i = 0; i < WINDOW\_SIZE && (base + i) < TOTAL\_PACKETS; ++i) {

int pkt = base + i;

if (isPacketLost()) {

cout << "Packet " << pkt << " lost. Timeout.\n";

lossOccurred = true;

break;

} else {

cout << "Packet " << pkt << " acknowledged.\n";

}

}

if (lossOccurred) {

cout << "Go-Back-N: Resending from packet " << base + i << "\n\n";

// Do not move base; resend from lost packet

} else {

// All packets in window acknowledged

base += i;

cout << " Window moved. New base = " << base << "\n\n";

}

}

cout << "All packets sent and acknowledged successfully.\n";

return 0;

}

**Output:**

Simulating Go-Back-N ARQ Protocol

------------------------------------

Total Packets : 10

Window Size : 4

Loss Probability : 20%

Sending window: [ 0 1 2 3 ]

Packet 0 acknowledged.

Packet 1 lost. Timeout.

Go-Back-N: Resending from packet 1

Sending window: [ 0 1 2 3 ]

Packet 0 lost. Timeout.

Go-Back-N: Resending from packet 0

Sending window: [ 0 1 2 3 ]

Packet 0 acknowledged.

Packet 1 acknowledged.

Packet 2 acknowledged.

Packet 3 acknowledged.

Window moved. New base = 4

Sending window: [ 4 5 6 7 ]

Packet 4 acknowledged.

Packet 5 acknowledged.

Packet 6 lost. Timeout.

Go-Back-N: Resending from packet 6

Sending window: [ 4 5 6 7 ]

Packet 4 acknowledged.

Packet 5 acknowledged.

Packet 6 lost. Timeout.

Go-Back-N: Resending from packet 6

Sending window: [ 4 5 6 7 ]

Packet 4 acknowledged.

Packet 5 acknowledged.

Packet 6 acknowledged.

Packet 7 acknowledged.

Window moved. New base = 8

Sending window: [ 8 9 ]

Packet 8 acknowledged.

Packet 9 acknowledged.

Window moved. New base = 10

All packets sent and acknowledged successfully.