

**Experiment No.:5**

| **TITLE:** Flow control Mechanism: Selective Repeat ARQ Sliding Window Protocol using Socket programming |
| --- |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**AIM:** Implementation of Flow Control Mechanism: Stop and Wait ARQ / Go-Back- N

/ Selective Repeat Sliding Window Protocol ARQ using sockets.

**Expected Outcome of Experiment:**

**CO:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. A. S. Tanenbaum, “Computer Networks”, Pearson Education, Fourth Edition
2. B. A. Forouzan, “Data Communications and Networking”, TMH, Fourth Edition

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

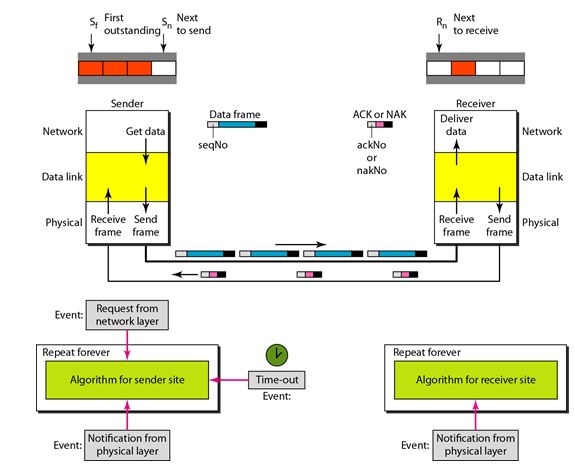
**Pre-Lab/ Prior Concepts:**

Java Socket Programming, Flow Control, Go-Back-Stop and Wait

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**New Concepts to be learned:** Window Flow Control **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Design of Go-Back-N ARQ**



1. Take data from user about how many bit windows is case of go back n and selective repeat.
2. Generate frames randomly and show the transmission
3. Generate the random number for the frame to be lost.
4. For Go – Back – N transmit all the frames after that number till max number
5. For Selective repeat transmit the selected frame which is not received by the receiver.

**IMPLEMENTATION: (**printout of code)

Client.py

import socket

import time

import random

WINDOW\_SIZE = 4

PACKET\_DROP\_PROBABILITY = 0.3

def client():

sender\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

sender\_socket.settimeout(2)

addr = ('localhost', 12345)

base = 0

next\_seq = 0

data = [str(i) for i in range(10)]

while base < len(data):

while next\_seq < base + WINDOW\_SIZE and next\_seq < len(data):

if random.random() > PACKET\_DROP\_PROBABILITY:

sender\_socket.sendto(data[next\_seq].encode(), addr)

print(f"Sent packet {next\_seq}")

else:

print(f"Dropped packet {next\_seq}")

next\_seq += 1

try:

ack, \_ = sender\_socket.recvfrom(1024)

ack = int(ack.decode())

print(f"Received ACK {ack}")

base = ack + 1

if base == next\_seq:

time.sleep(1)

except socket.timeout:

print("Timeout, resending window")

next\_seq = base

if \_\_name\_\_ == "\_\_main\_\_":

client()

server.py

import socket

import random

ACK\_DROP\_PROBABILITY = 0.3 # 30% chance of dropping ACKs

def server():

receiver\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

receiver\_socket.bind(('localhost', 12345))

expected\_seq = 0

while True:

data, addr = receiver\_socket.recvfrom(1024)

seq = int(data.decode())

if seq == expected\_seq:

print(f"Received packet {seq}")

if random.random() > ACK\_DROP\_PROBABILITY:

receiver\_socket.sendto(str(seq).encode(), addr)

print(f"Sent ACK {seq}")

else:

print(f"Dropped ACK {seq}")

expected\_seq += 1

else:

if random.random() > ACK\_DROP\_PROBABILITY:

receiver\_socket.sendto(str(expected\_seq - 1).encode(), addr)

print(f"Sent duplicate ACK {expected\_seq - 1}")

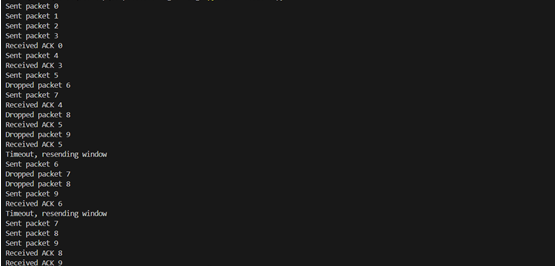
else:

print(f"Dropped duplicate ACK {expected\_seq - 1}")

if \_\_name\_\_ == "\_\_main\_\_":

server()





**CONCLUSION:**Learned go back n arq in socket programming.

**Post Lab Questions**

1. Compare Go-Back-N and Stop and Wait.  
     
   Stop-and-Wait ARQ: This protocol sends one frame at a time and waits for an acknowledgment (ACK) before sending the next frame. If the ACK is not received within a certain timeout period, the sender retransmits the frame.

* Pros: Simple and easy to implements
* Cons: Low efficiency for high-latency networks as it waits for an ACK after each frame, resulting in idle time.

Go-Back-N ARQ: This protocol allows the sender to send multiple frames within a specified window size before needing an ACK. If an error occurs in one frame, all subsequent frames are retransmitted ( "go back").

* Pros: Higher efficiency in terms of network utilization as multiple frames can be sent without waiting for an ACK for each one.
* Cons: If an error occurs, the protocol retransmits a large number of frames, even if only one was lost, which can lead to inefficiency

1. What is Flow Control and why it is necessary?  
     
   Flow control is a mechanism that manages the rate of data transmission between a sender and a receiver to prevent overwhelming the receiver. It ensures that the sender does not send data faster than the receiver can process it. Flow control is essential because it prevents buffer overflow at the receiver’s end, avoids data loss, and ensures efficient utilization of network resources.

1. The maximum window size for data transmission using the selective reject protocol with n-bit frame sequence numbers is  
   a) 2n            b) 2n-1                  **c) 2n-1**               d)2n-2

**Date: 30/10/24 Signature of Faculty In-charge**