**Batch: A2 Roll No.: 16010121045**

**Experiment / assignment / tutorial No.\_\_\_\_\_\_\_**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

**Experiment No.:5**

|  |
| --- |
| **TITLE:** Flow control Mechanism: Go-Back- N ARQ Sliding Window Protocol using Socket programming |

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

**AIM:** Implementation of Flow Control Mechanism: Stop and Wait ARQ and Go-Back- N Sliding Window Protocol ARQ using sockets.

**Expected Outcome of Experiment:**

**CO: Demonstrate Transport layer concepts like flow control, error control, congestion, sockets, QoS in wired technology.**

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

**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

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

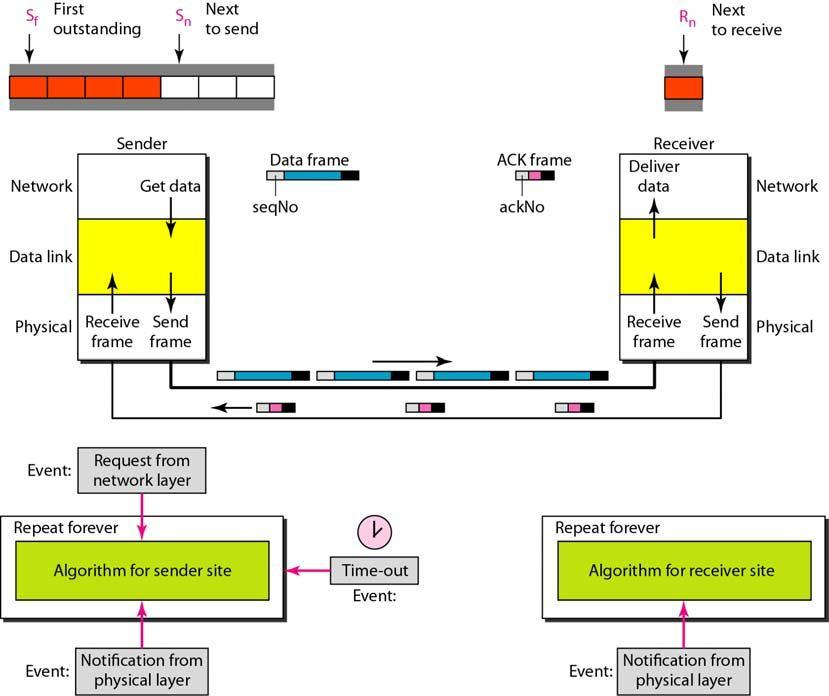
**Pre-Lab/ Prior Concepts:**

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)

Sender.py

*import* socket

*import* time

*# Create a socket*

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

host = "localhost" *# Use the appropriate host address*

port = 12345 *# Use the appropriate port number*

*# Bind the socket to the address*

server\_socket.bind((host, port))

*# Listen for incoming connections*

server\_socket.listen(5)

print("Waiting for a connection...")

conn, addr = server\_socket.accept()

print("Connection established with", addr)

*# Receive the client's name*

s\_name = conn.recv(1024)

s\_name = s\_name.decode()

print(s\_name, "has connected to the chat room")

conn.send(s\_name.encode())

*while* True:

message = input("Me: ")

*# Send the message to the client*

conn.send(message.encode())

*if* message == "[e]":

message = "Left chat room!"

conn.send(message.encode())

print("\nConnection closed")

*break*

*# Assuming 'binarycode' is a valid function to encode messages*

*# message = binarycode(message*

*# Get the length of the message*

message\_length = str(len(message))

conn.send(message\_length.encode())

i = 0

j = 0

j = int(input("Enter the window size: "))

conn.send(str(j).encode())

f = int(message\_length)

k = j

*while* i != f:

print("Sending frames range:", i, "to", min(i + j, f))

*for* fr *in* range(i, min(i + j, f)):

conn.send(str(fr).encode())

print("Frame sent:", fr)

time.sleep(0.5)

i = int(conn.recv(1024).decode())

print("Received cumulative acknowledgement:", i)

*# Close the connection and the server socket*

conn.close()

server\_socket.close()

**reciver.py**

*import* time

*import* socket

*import* sys

*import* random

print("\nWelcome to the chat room\n")

print("\*Initializing\*\n")

time.sleep(1)

s = socket.socket()

shost = socket.gethostname()

ip = socket.gethostbyname(shost)

print(shost, "(", ip, ")\n")

host = ip

name = "receiver"

port = 1235

print("\nTrying to connect to", host, "(", port, ")\n")

time.sleep(1)

s.connect((host, port))

print("Connected...\n")

s.send(name.encode())

s\_name = s.recv(1024)

s\_name = s\_name.decode()

print(s\_name, "has joined the chat room\nEnter [e] to exit the chat room\n")

*while* True:

m = s.recv(1024)

m = m.decode()

k = s.recv(1024)

k = k.decode()

k = int(k)

ws = int(s.recv(1024).decode())

i = 0

a = ""

b = ""

f = random.randint(0, 1)

message = ""

*while* i != k:

t = i

*for* \_ *in* range(t, min(t + ws, k)):

r = int(s.recv(1024).decode())

f = random.randint(0, 1)

*if* f:

print("Frame lost:", r)

*continue*

*if* r == i:

print("Frame received:", r)

i += 1

*else*:

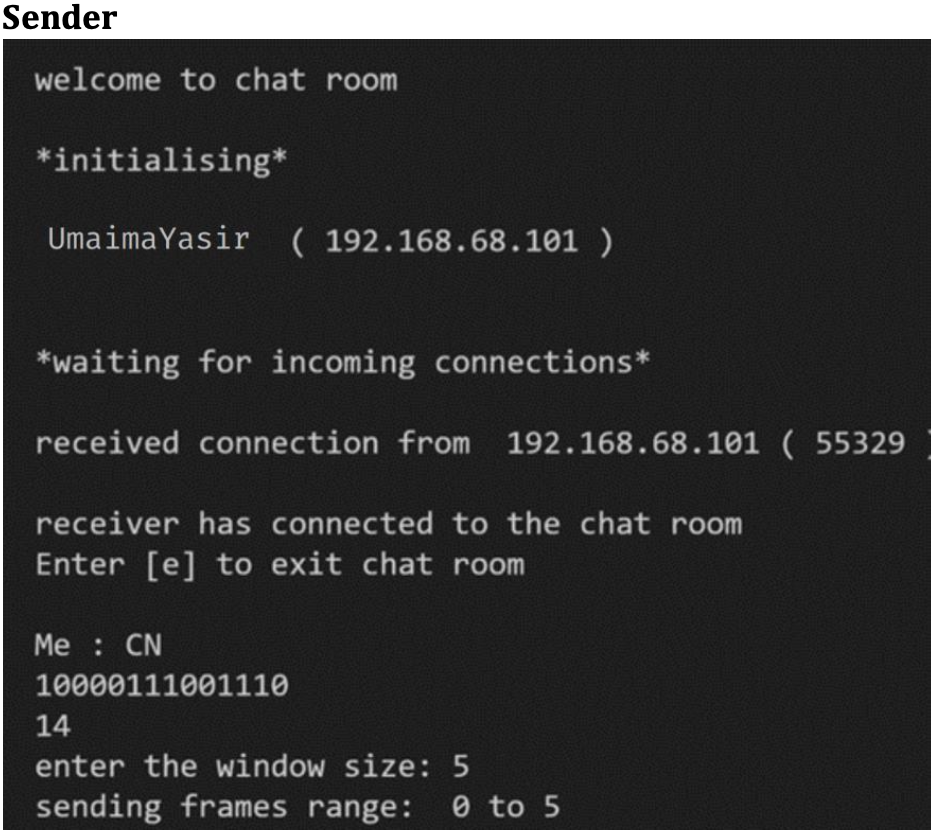
print("Discarded frame:", r)

s.send(str(i).encode())

print("Sent cumulative acknowledgement:", i)

print("The message received is:", m)

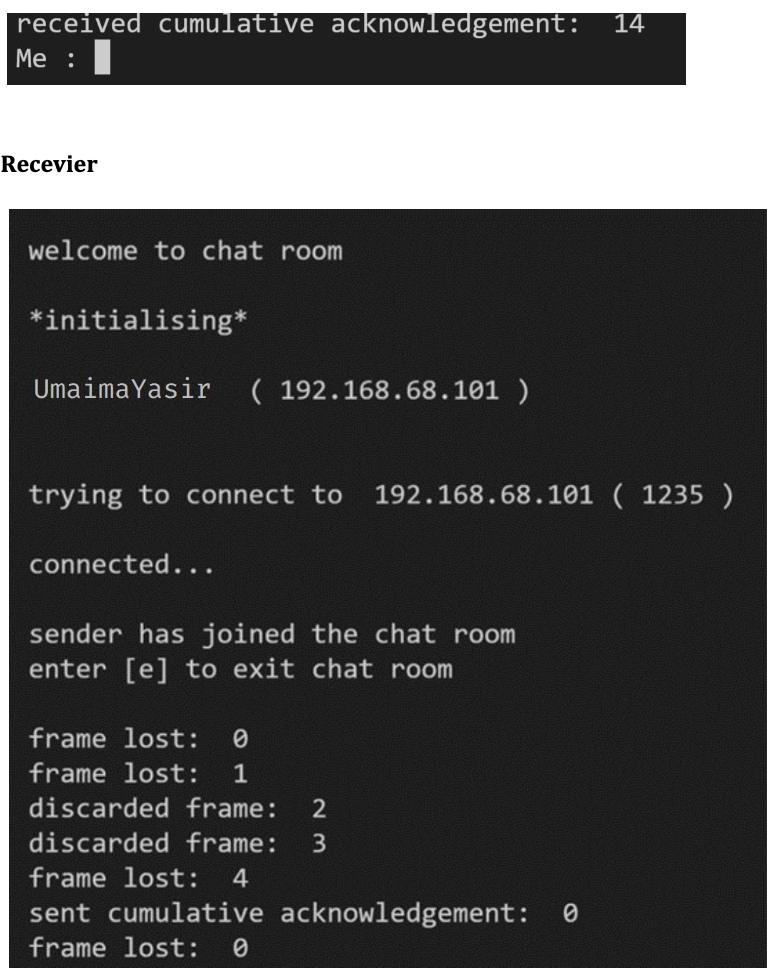
**Output**

****

**A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated**

****

**A screen shot of a computer

Description automatically generatedA screen shot of a computer

Description automatically generatedCONCLUSION:** Hence, we have successfully implemented the Flow Control Mechanism: Stop and Wait ARQ and Go-Back- N Sliding Window Protocol ARQ.

**Post Lab Questions**

1. Compare Go-Back-N and Stop and Wait.

A table with text on it

Description automatically generated

1. What is Flow Control and why it is necessary?

Flow control is design issue at Data Link Layer. It is technique that generally observes proper flow of data from sender to receiver. It is very essential because it is possible for sender to transmit data or information at very fast rate and hence receiver can receive this information and process it. This can happen only if receiver has very high load of traffic as compared to sender, or if receiver has power of processing less as compared to sender.

Flow control is basically technique that gives permission to two of stations that are working and processing at different speeds to just communicate with one another. Flow control in Data Link Layer simply restricts and coordinates number of frames or amount of data sender can send just before it waits for an acknowledgment from receiver. Flow control is actually set of procedures that explains sender about how much data or frames it can transfer or transmit before data overwhelms receiver.

The receiving device also contains only limited amount of speed and memory to store data. This is why receiving device should be able to tell or inform the sender about stopping the transmission or transferring of data on temporary basis before it reaches limit. It also needs buffer, large block of memory for just storing data or frames until they are processed.

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 :\_\_\_\_\_\_\_\_\_\_ Signature of Faculty In-charge**