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

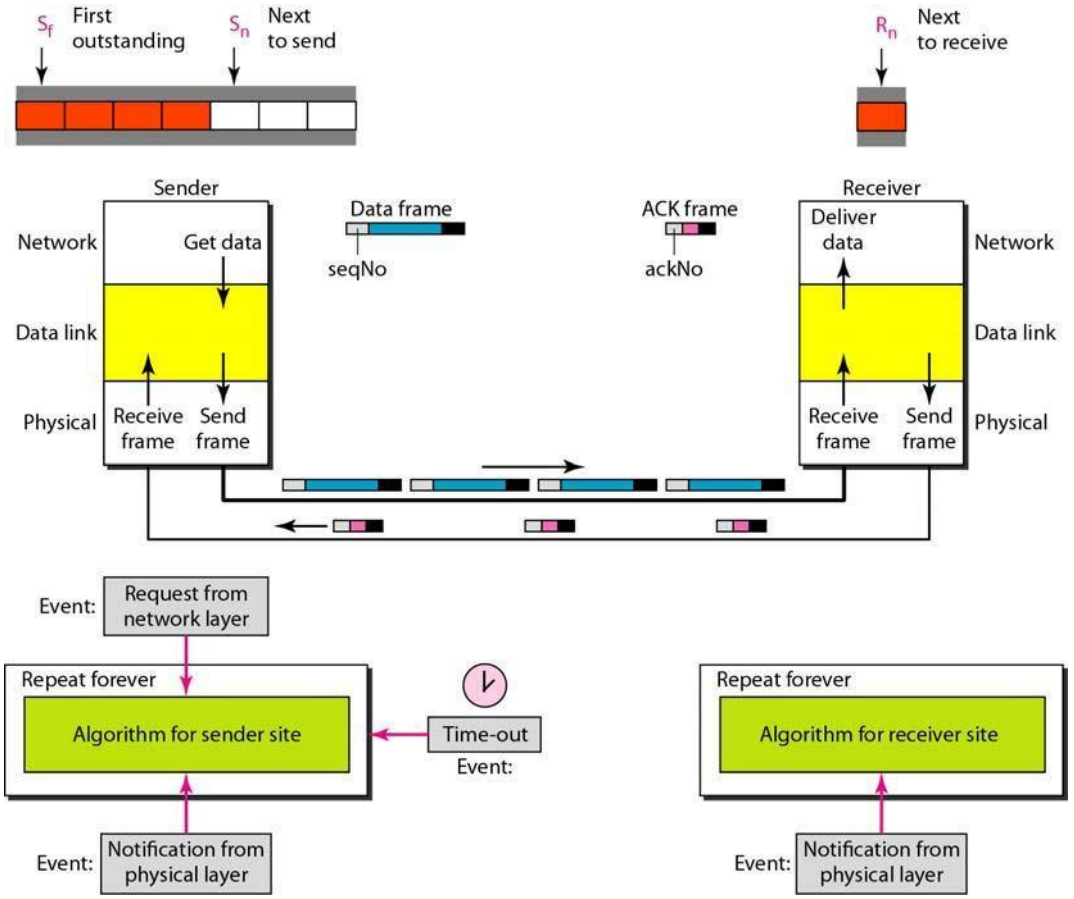
---

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

Sender

```
welcome to chat room

*initialising*

 UmaimaYasir  ( 192.168.68.101 )


*waiting for incoming connections*

received connection from  192.168.68.101 ( 55329 )

receiver has connected to the chat room
Enter [e] to exit chat room

Me : CN
10000111001110
14
enter the window size: 5
sending frames range:  0 to 5
```

```
frame sent:  0
frame sent:  1
frame sent:  2
frame sent:  3
frame sent:  4
received cumulative acknowledgement:  0
sending frames range:  0 to 5
frame sent:  0
frame sent:  1
frame sent:  2
frame sent:  3
frame sent:  4
received cumulative acknowledgement:  0
sending frames range:  0 to 5
frame sent:  0
frame sent:  1
frame sent:  2
frame sent:  3
frame sent:  4
received cumulative acknowledgement:  0
sending frames range:  0 to 5
frame sent:  0
frame sent:  1
frame sent:  2
frame sent:  3
frame sent:  4
```

```
received cumulative acknowledgement:  0
sending frames range:  0 to 5
frame sent:  0
frame sent:  1
frame sent:  2
frame sent:  3
frame sent:  4
received cumulative acknowledgement:  2
sending frames range:  2 to 7
frame sent:  2
frame sent:  3
frame sent:  4
frame sent:  5
frame sent:  6
received cumulative acknowledgement:  5
sending frames range:  5 to 10
frame sent:  5
frame sent:  6
frame sent:  7
frame sent:  8
frame sent:  9
received cumulative acknowledgement:  5
sending frames range:  5 to 10
frame sent:  5
frame sent:  6
frame sent:  7
```

```
frame sent:  8
frame sent:  9
received cumulative acknowledgement:  8
sending frames range:  8 to 13
frame sent:  8
frame sent:  9
frame sent:  10
frame sent:  11
frame sent:  12
received cumulative acknowledgement:  8
sending frames range:  8 to 13
frame sent:  8
frame sent:  9
frame sent:  10
frame sent:  11
frame sent:  12
received cumulative acknowledgement:  9
sending frames range:  9 to 14
frame sent:  9
frame sent:  10
frame sent:  11
frame sent:  12
frame sent:  13
received cumulative acknowledgement:  10
sending frames range:  10 to 14
frame sent:  10
frame sent:  11
frame sent:  12
frame sent:  13
received cumulative acknowledgement:  10
sending frames range:  10 to 14
frame sent:  10
frame sent:  11
frame sent:  12
frame sent:  13
received cumulative acknowledgement:  11
sending frames range:  11 to 14
frame sent:  11
frame sent:  12
frame sent:  13
received cumulative acknowledgement:  13
sending frames range:  13 to 14
frame sent:  13
received cumulative acknowledgement:  13
sending frames range:  13 to 14
frame sent:  13
received cumulative acknowledgement:  13
sending frames range:  13 to 14
frame sent:  13
```

```
received cumulative acknowledgement:  14
Me :
```

Recevier

```
welcome to chat room

*initialising*

UmaimaYasir  ( 192.168.68.101 )


trying to connect to  192.168.68.101 ( 1235 )

connected...

sender has joined the chat room
enter [e] to exit chat room

frame lost:  0
frame lost:  1
discarded frame:  2
discarded frame:  3
frame lost:  4
sent cumulative acknowledgement:  0
frame lost:  0
```

```
discarded frame:  1
frame lost:  2
discarded frame:  3
discarded frame:  4
sent cumulative acknowledgement:  0
frame lost:  0
frame lost:  1
discarded frame:  2
frame lost:  3
frame lost:  4
sent cumulative acknowledgement:  0
frame lost:  0
discarded frame:  1
discarded frame:  2
discarded frame:  3
frame lost:  4
sent cumulative acknowledgement:  0
frame received:  0
frame received:  1
frame lost:  2
discarded frame:  3
discarded frame:  4
sent cumulative acknowledgement:  2
frame received:  2
frame received:  3
frame received:  4
```

```
frame lost:  5
frame lost:  6
sent cumulative acknowledgement:  5
frame lost:  5
discarded frame:  6
discarded frame:  7
discarded frame:  8
discarded frame:  9
sent cumulative acknowledgement:  5
frame received:  5
frame received:  6
frame received:  7
frame lost:  8
frame lost:  9
sent cumulative acknowledgement:  8
frame lost:  8
discarded frame:  9
discarded frame:  10
discarded frame:  11
frame lost:  12
sent cumulative acknowledgement:  8
frame received:  8
frame lost:  9
frame lost:  10
discarded frame:  11
discarded frame:  12
```

```
sent cumulative acknowledgement:  9
frame received:  9
frame lost:  10
frame lost:  11
frame lost:  12
frame lost:  13
sent cumulative acknowledgement:  10
frame lost:  10
frame lost:  11
discarded frame:  12
discarded frame:  13
sent cumulative acknowledgement:  10
frame received:  10
frame lost:  11
discarded frame:  12
discarded frame:  13
sent cumulative acknowledgement:  11
frame received:  11
frame received:  12
frame lost:  13
sent cumulative acknowledgement:  13
frame lost:  13
sent cumulative acknowledgement:  13
frame lost:  13
sent cumulative acknowledgement:  13
frame lost:  13
sent cumulative acknowledgement:  13
frame received:  13
sent cumulative acknowledgement:  14
the message received is : CN
```

CONCLUSION: 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.

Sr. No.	Key	Stop and Wait protocol	GoBackN protocol
1	Sender window size	In Stop and Wait protocol, Sender window size is 1.	In GoBackN protocol, Sender window size is N.
2	Receiver Window size	In Stop and Wait protocol, Receiver window size is 1.	In GoBackN protocol, Receiver window size is 1.
3	Minimum Sequence Number	In Stop and Wait protocol, Minimum Sequence Number is 2.	In GoBackN protocol, Minimum Sequence Number is N+1 where N is number of packets sent.
4	Efficiency	In Stop and Wait protocol, Efficiency formular is 1/(1+2*a) where a is ratio of propagation delay vs transmission delay.	In GoBackN protocol, Efficiency formular is N/(1+2*a) where a is ratio of propagation delay vs transmission delay and N is number of packets sent.

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

Flow control is design issue at Data Link Layer. It is technique that generally observes properflow 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 informationand 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 DataLink 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.

3. 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