

## CN Experiment 4

Aim: To implement sliding window protocols - Selective Repeat and Go Back N.

### Theory:

Go Back N and Selective Repeat are two types of sliding window protocols used in data communication to ensure reliable transmission over unreliable or noisy communication channels. Both are part of the ARQ family, which are mechanisms that use acknowledgements and timeouts to achieve reliable data transmission.

1. Go-Back-N is a sliding window protocol that allows the sender to send multiple frames before needing an ACK for the first time frame, but the receiver is only required to remember the sequence number of the next frame it expects. The sender's window size,  $N$ , determines the maximum number of frames that can be sent without waiting for an acknowledgement. If sender does not receive an acknowledgement for a frame within a certain timeout period, it goes back and retransmits all frames starting from the unacknowledged frame. This reduces the efficiency if the network is unreliable.
2. Selective Repeat is used for error detection and control in data link layer. The sender sends several frames

specified by a window size even without the need to wait for individual acknowledgment from the receiver. The retransmitted frame is received out of sequence. Only the lost or error frames are retransmitted, which is an advantage over Go-Back-N.

Conclusion: We have studied and implemented various sliding window protocols.



Department of Computer Engineering  
Class: T.Y.B.Tech. Semester: V

Course Code: DJS22CEL502

Course Name: Computer Networks Lab

Name: Vruddhi Shah	SAP ID: 60004220215
Date of Performance: 28.08.24	Date of Submission: 29.08.24

## Experiment No: 5

**Aim:** Write a program to implement Framing:

Techniques: Character count, Byte stuffing, Bit stuffing.

**Program for GoBackN:**

```
sender.py
import socket
import time
import select
host = "127.0.0.1"
port = 10000
cs = socket.socket()
cs.connect((host, port))
cs.settimeout(2)
x = input("Enter string")
n = len(x)
i = 0
timeout = 10
lost = []
while i < n:
    start = time.time()
    while time.time()-start < timeout and i < n:
        cs.send((x[i] + "," + str(i)).encode())
        try:
            ack = cs.recv(1024).decode()

            if int(ack) == i + 1:
                print("Received ACK" + ack)
        except:
            lost.append(i)
        i += 1

    time.sleep(1)
    if len(lost) == 0:
        print("Nothing was lost")
    else:
        print("Something was lost. Starting from index", lost[0])
        i = lost[0]
```





Department of Computer Engineering  
 Class: T.Y.B.Tech. Semester: V

Course Code: DJS22CEL502

Course Name: Computer Networks Lab

```
lost = []
print("Full message transmitted")
cs.close()
```

**receiver.py**

```
import socket
import random
host = "127.0.0.1"
port = 10000
ss = socket.socket()
ss.bind((host, port))
ss.listen()
print("Established connection")
conn, address = ss.accept()
while True:
    got = (conn.recv(1024)).decode()
    if got=="":
        break
    print("Got frame", got[-1])
    i = int(got[-1])
    ack = str(i + 1)
    if random.randint(0, 1) == 1:
        print("Sending", ack)
        conn.send(ack.encode())
conn.close()
```

```
→ CN git:(main) ✗ cd GoBackN
→ GoBackN git:(main) ✗ python3 sender.py
Enter stringabcd
Received ACK1
Received ACK3
Received ACK4
Something was lost. Starting from index 1
Received ACK3
Something was lost. Starting from index 1
Received ACK4
Something was lost. Starting from index 1
Received ACK2
Received ACK4
Something was lost. Starting from index 2
Received ACK3
Received ACK4
Nothing was lost
Full message transmitted
```



Department of Computer Engineering  
Class: T.Y.B.Tech. Semester: V

Course Code: DJS22CEL502

Course Name: Computer Networks Lab

```
➔ GoBackN git:(main) x python3 receiver.py
```

```
Established connection
```

```
Got frame 0
```

```
Sending 1
```

```
Got frame 1
```

```
Got frame 2
```

```
Sending 3
```

```
Got frame 3
```

```
Sending 4
```

```
Got frame 1
```

```
Got frame 2
```

```
Sending 3
```

```
Got frame 3
```

```
Got frame 1
```

```
Got frame 2
```

```
Got frame 3
```

```
Sending 4
```

```
Got frame 1
```

```
Sending 2
```

```
Got frame 2
```

```
Got frame 3
```

```
Sending 4
```

```
Got frame 2
```

```
Sending 3
```

```
Got frame 3
```

```
Sending 4
```

**Program for Selective repeat:**

**sender.py**

```
import socket
```

```
import time
```

```
host = "127.0.0.1"
```

```
port = 10000
```

```
cs = socket.socket()
```

```
cs.connect((host, port))
```

```
cs.settimeout(2)
```

```
x = input("Enter string: ")
```

```
n = len(x)
```



**Department of Computer Engineering**  
**Class: T.Y.B.Tech. Semester: V**

**Course Code: DJS22CEL502**

**Course Name: Computer Networks Lab**

```
i = 0
ackn = [0] * n

cs.send(str(n).encode())
print("Sent n")

lost = []
timeout = 5

while i < n:
    start = time.time()

    while time.time() - start < timeout and i < n:
        cs.send((x[i] + "," + str(i)).encode())
        print("Sending", x[i] + "," + str(i))

        try:
            ack = cs.recv(1024).decode()
            arr = ack.split(",")

            for a in arr:
                if a.isdigit() and int(a) - 1 < n:
                    ackn[int(a) - 1] = 1
                    print("Got acknowledgment", a)
            if len(arr) == 0:
                lost.append(i)

        except socket.timeout:
            continue

    i += 1
    time.sleep(1)

if len(lost) == 0:
    print("All frames received successfully")
    break
else:
    print("Some ack lost")
    first_lost = lost[0]
    lost = []

    for j in range(first_lost, n):
        if ackn[j] == 1:
            print("Resending acknowledgment for frame", j + 1)
            cs.send((str(j + 1) + ",").encode())
            time.sleep(0.5)
```



**Department of Computer Engineering**  
**Class: T.Y.B.Tech. Semester: V**

**Course Code: DJS22CEL502**

**Course Name: Computer Networks Lab**

```
print("Full message transmitted")
cs.close()
receiver.py
import socket
import random

host = "127.0.0.1"
port = 10000

ss = socket.socket()
ss.bind((host, port))
ss.listen()
print("Established connection")

conn, address = ss.accept()

n = int(conn.recv(1024).decode())
print(n)
received = [0] * n

while True:
    got = conn.recv(1024).decode()
    if not got:
        break

    frame_number = int(got.split(",")[1])
    print("Got frame", frame_number)

    if received[frame_number] == 0:
        received[frame_number] = 1
        ack = str(frame_number + 1)
        print("Frame", frame_number, "received correctly. Sending ACK", ack)
    else:
        print("Duplicate frame", frame_number)
        ack_list = [str(i + 1) for i in range(frame_number, n) if received[i]]
        ack = ",".join(ack_list)
        print("Resending ACKs for frames", ack_list)

    if random.randint(0, 1) == 1:
        conn.send(ack.encode())
        print("Sent ACK", ack)
    else:
        print("Simulated ACK loss")

conn.close
```



Department of Computer Engineering  
Class: T.Y.B.Tech. Semester: V

Course Code: DJS22CEL502

Course Name: Computer Networks Lab

```
● → CN git:(main) x /usr/bin/python3 /Users/v
Enter string: abc
Sent n
Sending a,0
Got acknowledgment 1
Sending b,1
Sending b,1
Got acknowledgment 2
Sending c,2
All frames received successfully
Full message transmitted

▶ → CN git:(main) x /usr/bin/python3 /Users/vrudan1/Des
Established connection
3
Got frame 0
Frame 0 received correctly. Sending ACK 1
Sent ACK 1
Got frame 1
Frame 1 received correctly. Sending ACK 2
Simulated ACK loss
Got frame 1
Duplicate frame 1
Resending ACKs for frames ['2']
Sent ACK 2
Got frame 2
Frame 2 received correctly. Sending ACK 3
Simulated ACK loss
```

**Conclusion:** We have successfully learnt and implemented framing.