

# Internet Protocols

By :

Abdelrahman Samha 2136821

Instructed by: Dr.Mohammad Al-Hammouri

This project implements two **reliable data transfer protocols**, **Go-Back-N (GBN)** and **Selective Repeat (SR)**, on top of the inherently unreliable **UDP protocol**. UDP does not guarantee delivery, ordering, or protection against duplication, so the goal of this project is to build a **reliability layer** that ensures correct and ordered delivery of data between a sender and a receiver.

## Key Features Implemented

### 1. Protocol Modes

- **Go-Back-N (GBN):** The sender retransmits all unacknowledged packets upon timeout. The receiver accepts only in-order packets and discards any out-of-order packets, sending cumulative acknowledgments (ACKs).
- **Selective Repeat (SR):** The sender retransmits only the packets that were lost. The receiver buffers out-of-order packets and delivers them in order, sending individual ACKs for each correctly received packet.

### 2. Sender Program

- Reads data from a file and segments it into fixed-size packets (512B).
- Implements a sliding window for packet transmission.
- Sends packets with sequence numbers and handles ACKs and retransmissions.
- Simulates packet loss and transmission delay for robustness testing.
- Logs all sent, acknowledged, and retransmitted packets with timestamps.

### 3. Receiver Program

- Receives packets, simulates packet loss and delay, and sends ACKs for correctly received packets.
- For GBN, only in-order packets are accepted.
- For SR, out-of-order packets are buffered and delivered when possible.
- Writes received data to an output file and logs all events with timestamps.

### 4. Robustness Features

- Configurable **loss rate** and **transmission delay** to simulate real-world unreliable networks.
- Graceful termination with "END" message from sender to signal end of transmission.
- Logging of all key events to monitor protocol behavior and packet flow.

## Testing Scenarios

- **Basic Test:** No loss or delay to verify correct transmission.
- **Loss Simulation:** Introduce packet loss to test retransmissions.
- **Delay Simulation:** Add delays to test sender timeout handling.
- **Mixed Conditions:** Combine loss and delay to compare GBN and SR efficiency.
- **Stress Test:** Transfer a large file with multiple packets and a sliding window to evaluate performance under load.

## Technical Implementation

- Language: Python
- Communication: UDP sockets
- User-configurable parameters: mode (GBN/SR), window size, loss rate, delay, file to send/receive
- Logging: Real-time display of sent, received, acknowledged, and retransmitted packets

## Outcome:

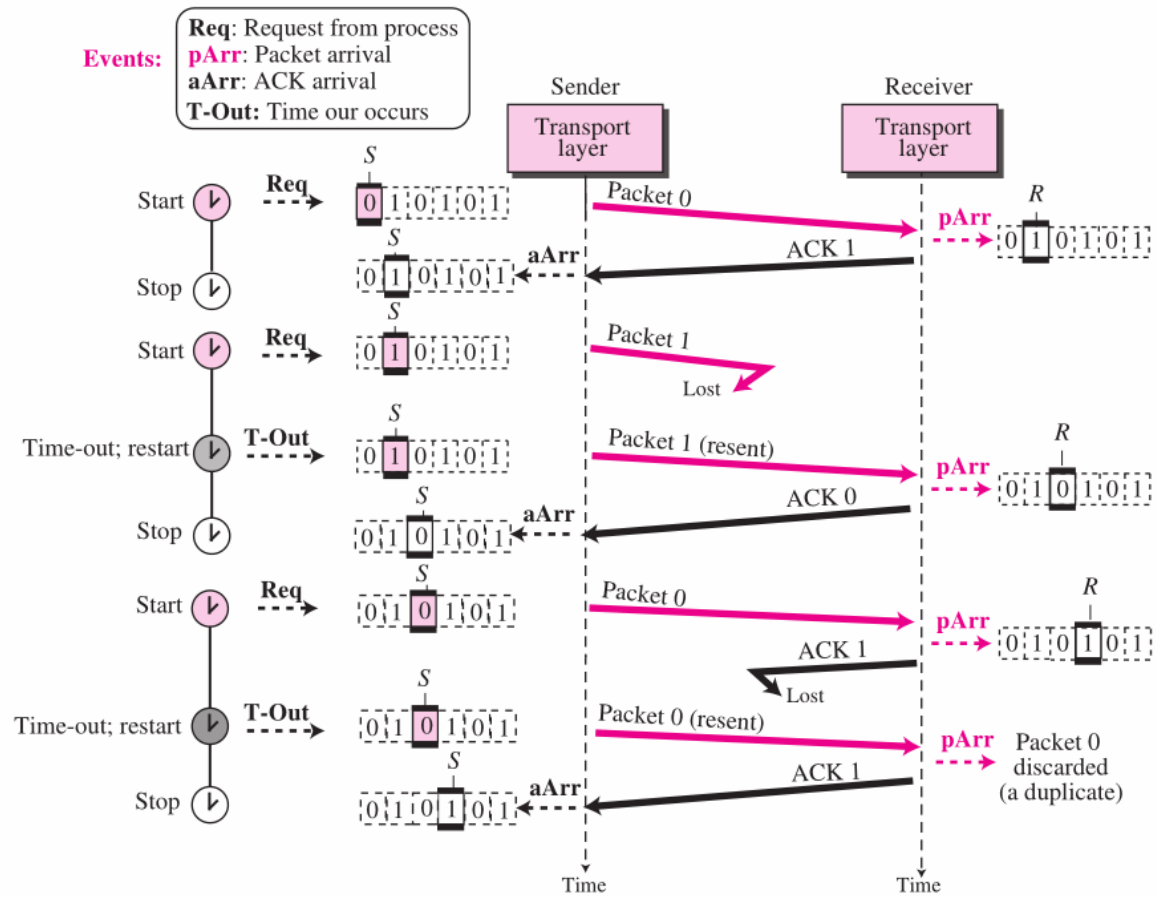
The project successfully demonstrates reliable data transfer over an unreliable transport protocol, implementing sliding-window mechanisms, ACK handling, retransmissions, and handling of packet loss, duplication, and out-of-order delivery.

## Simulation and Testing

In this section we will cover the 5 test cases for each operating mode (GBN/ SR) as requested, the 5 test cases are :

- 1- Basic Test (no loss, no delay): Ensure correct file transfer.
- 2- Loss Simulation: Test with loss rate 0.1, 0.3 and verify retransmissions.
- 3- Delay Simulation: Delay 0.2s to test timeout handling.
- 4- Mixed Conditions: Loss rate 0.2 + delay 0.1s; compare GBN vs SR.
- 5- Stress Test: Large file (≥100 packets), window size = 10.

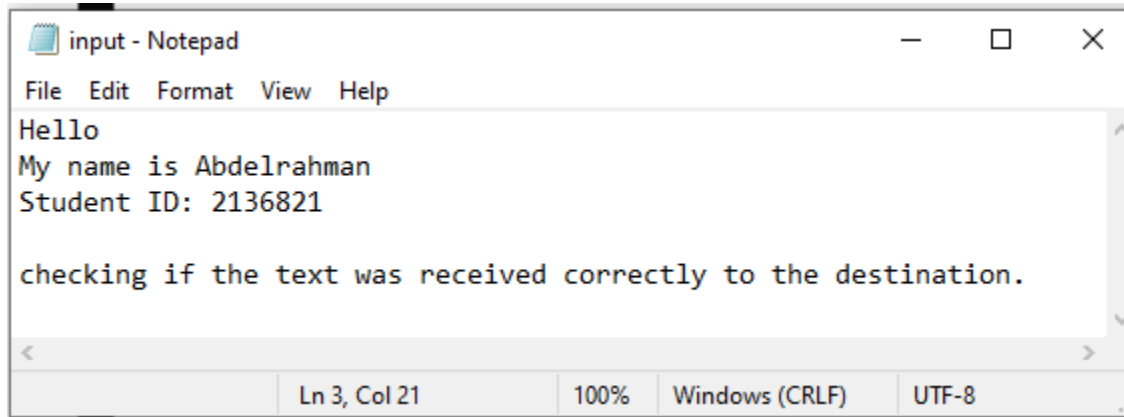
the following diagram from the TCP/IP protocol suite by B.Forouzan shows how the GBN protocol works to insure correctness in delivery of UDP packets.



## GBN TESTS

### 1 – BASIC TEST

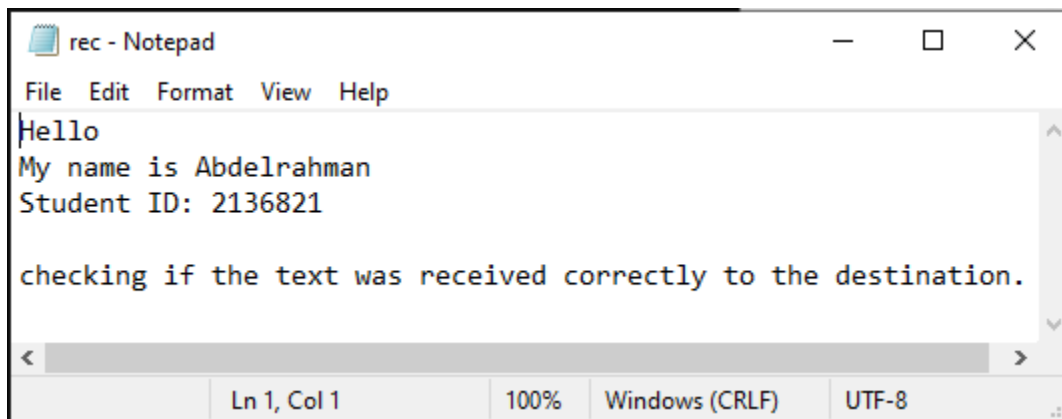
Input data :



```
input - Notepad
File Edit Format View Help
Hello
My name is Abdelrahman
Student ID: 2136821

checking if the text was received correctly to the destination.
```

Received data :



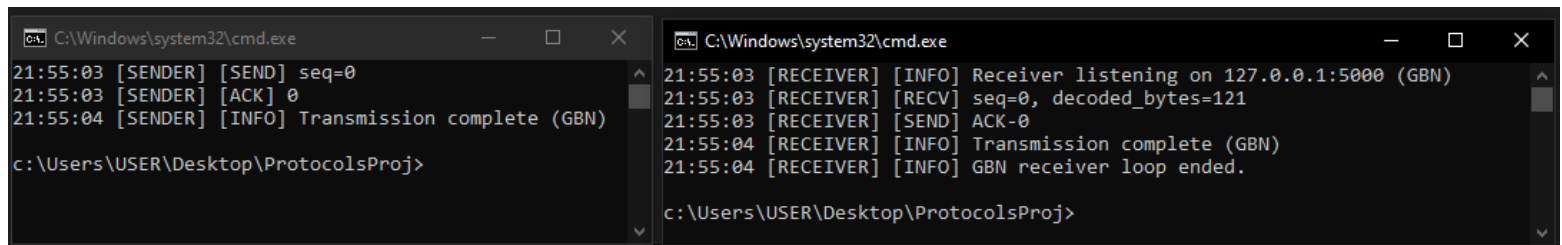
```
rec - Notepad
File Edit Format View Help
Hello
My name is Abdelrahman
Student ID: 2136821

checking if the text was received correctly to the destination.
```

Data transmission :

Sender :

Receiver:



```
C:\Windows\system32\cmd.exe
21:55:03 [SENDER] [SEND] seq=0
21:55:03 [SENDER] [ACK] 0
21:55:04 [SENDER] [INFO] Transmission complete (GBN)
c:\Users\USER\Desktop\ProtocolsProj>

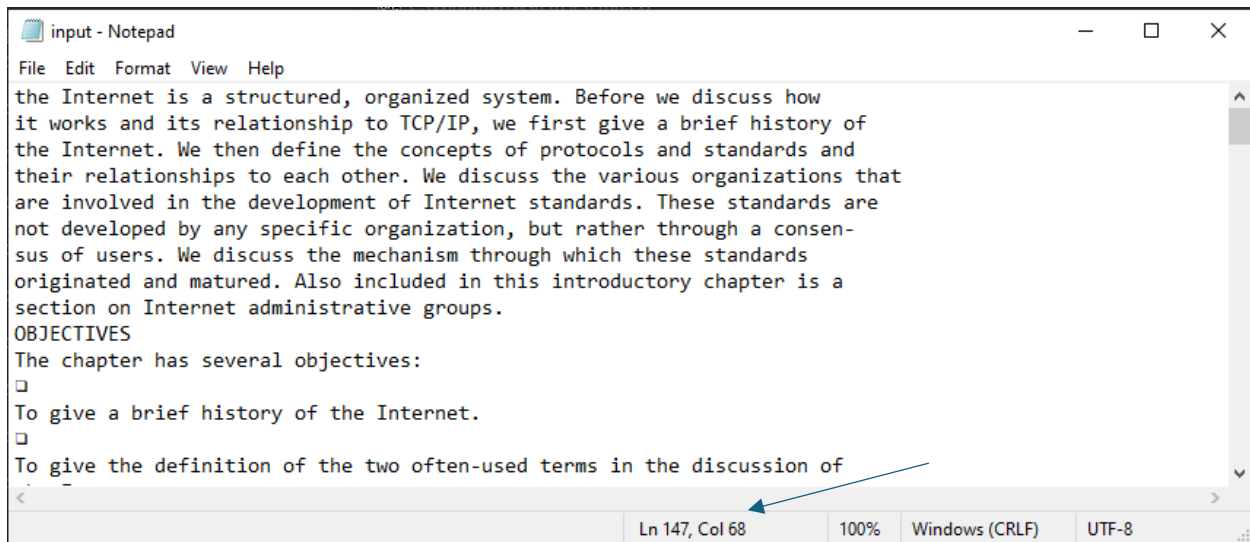
C:\Windows\system32\cmd.exe
21:55:03 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
21:55:03 [RECEIVER] [RECV] seq=0, decoded_bytes=121
21:55:03 [RECEIVER] [SEND] ACK-0
21:55:04 [RECEIVER] [INFO] Transmission complete (GBN)
21:55:04 [RECEIVER] [INFO] GBN receiver loop ended.
c:\Users\USER\Desktop\ProtocolsProj>
```

The data is sent in a single packet with the size of 121, this is because the file is very small, and the program segments the data 512Bytes at a time, but if its smaller then it just takes the exact same size.

## 2- LOSS SIMULATION :

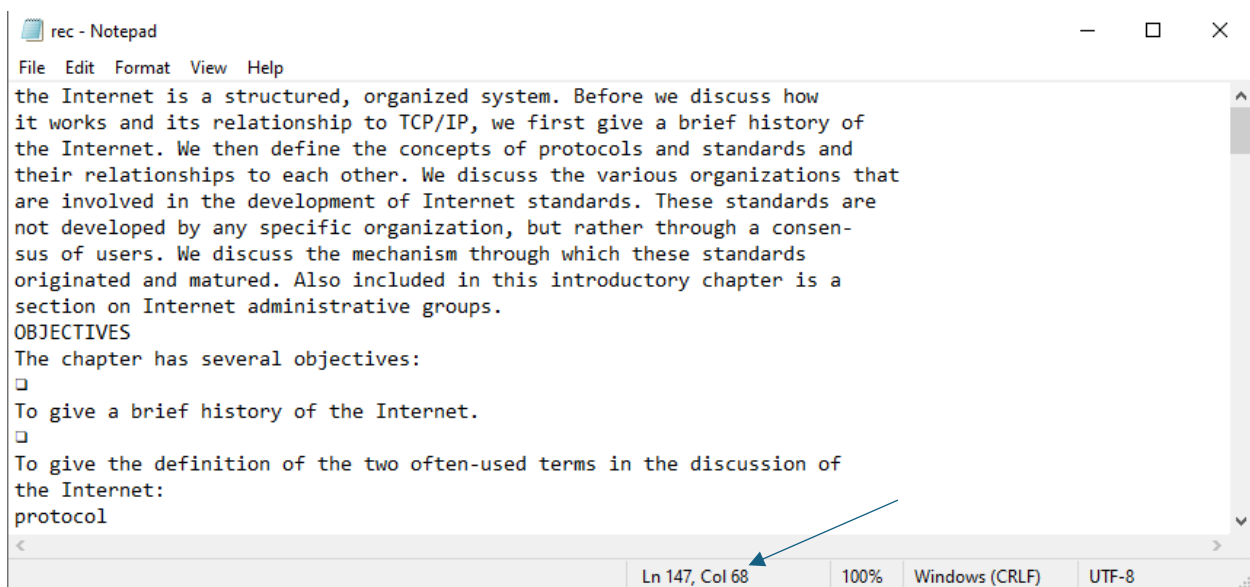
A small text wont help in this, because we need to have a large file to be able to send multiple packets to simulate some of them getting dropped.

For this test I took the introduction section of the book TCP/IP protocol suite by Forouzan to be able to send 10+ packets.



The ln 147, Col 68 shows the size of the program, because I placed my cursor at the last character of the text being sent. This is to proof that the transmitted file is of the same size.

The received file :



This test had the packet loss set to 0.1, we had some packets lost in the process :

```
C:\Windows\system32\cmd.exe
22:01:35 [SENDER] [SEND] seq=0
22:01:35 [SENDER] [SEND] seq=1
22:01:35 [SENDER] [SEND] seq=2
22:01:35 [SENDER] [SEND] seq=3
22:01:35 [SENDER] [ACK] 0
22:01:35 [SENDER] [SEND] seq=4
22:01:35 [SENDER] [ACK] 1
22:01:35 [SENDER] [SEND] seq=5
22:01:35 [SENDER] [ACK] 2
22:01:35 [SENDER] [SEND] seq=6
22:01:35 [SENDER] [ACK] 3
22:01:35 [SENDER] [SEND] seq=7
22:01:35 [SENDER] [ACK] 4
22:01:35 [SENDER] [SEND] seq=8
22:01:35 [SENDER] [ACK] 5
22:01:35 [SENDER] [SEND] seq=9
22:01:35 [SENDER] [ACK] 6
22:01:35 [SENDER] [LOSS] Dropped packet: DATA-10-aWQgbXVjaCBm
22:01:35 [SENDER] [SEND] seq=10
22:01:35 [SENDER] [SEND] seq=11
22:01:35 [SENDER] [SEND] seq=12
22:01:35 [SENDER] [ACK] 9
22:01:35 [SENDER] [SEND] seq=13
22:01:35 [SENDER] [ACK] 9
22:01:35 [SENDER] [ACK] 9
22:01:37 [SENDER] [TIMEOUT] Resending window
22:01:37 [SENDER] [RETX] seq=10
22:01:37 [SENDER] [RETX] seq=11
22:01:37 [SENDER] [RETX] seq=12
22:01:37 [SENDER] [RETX] seq=13
22:01:37 [SENDER] [ACK] 10
22:01:37 [SENDER] [ACK] 11
22:01:37 [SENDER] [ACK] 12
22:01:37 [SENDER] [ACK] 13
22:01:37 [SENDER] [INFO] Transmission complete (GBN)

c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
22:01:35 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
22:01:35 [RECEIVER] [RECV] seq=0, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-0
22:01:35 [RECEIVER] [RECV] seq=1, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-1
22:01:35 [RECEIVER] [RECV] seq=2, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-2
22:01:35 [RECEIVER] [RECV] seq=3, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-3
22:01:35 [RECEIVER] [RECV] seq=4, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-4
22:01:35 [RECEIVER] [RECV] seq=5, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-5
22:01:35 [RECEIVER] [RECV] seq=6, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-6
22:01:35 [RECEIVER] [RECV] seq=7, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-7
22:01:35 [RECEIVER] [RECV] seq=8, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-8
22:01:35 [RECEIVER] [RECV] seq=9, decoded_bytes=512
22:01:35 [RECEIVER] [SEND] ACK-9
22:01:35 [RECEIVER] [SEND] ACK-9 (dup)
22:01:35 [RECEIVER] [SEND] ACK-9 (dup)
22:01:35 [RECEIVER] [SEND] ACK-9 (dup)
22:01:37 [RECEIVER] [RECV] seq=10, decoded_bytes=512
22:01:37 [RECEIVER] [SEND] ACK-10
22:01:37 [RECEIVER] [RECV] seq=11, decoded_bytes=512
22:01:37 [RECEIVER] [SEND] ACK-11
22:01:37 [RECEIVER] [RECV] seq=12, decoded_bytes=512
22:01:37 [RECEIVER] [SEND] ACK-12
22:01:37 [RECEIVER] [RECV] seq=13, decoded_bytes=336
22:01:37 [RECEIVER] [SEND] ACK-13
22:01:37 [RECEIVER] [INFO] Transmission complete (GBN)
22:01:37 [RECEIVER] [INFO] GBN receiver loop ended.

c:\Users\USER\Desktop\ProtocolsProj>
```

Packet 10 was dropped and was not sent to the receiver, this made the receiver send ACK9 to indicate that the last received packet was 9 and that it did not receive packet 10, so the sender retransmitted the packet 10 later on along with all the packets that came after it.

Now testing with 0.3 packet loss with the same file:

```
C:\Windows\system32\cmd.exe
22:11:36 [SENDER] [SEND] seq=0
22:11:36 [SENDER] [LOSS] Dropped packet: DATA-1-YXJkcw0K
22:11:36 [SENDER] [SEND] seq=1
22:11:36 [SENDER] [LOSS] Dropped packet: DATA-2-aA0Kd2hp
22:11:36 [SENDER] [SEND] seq=2
22:11:36 [SENDER] [SEND] seq=3
22:11:36 [SENDER] [ACK] 0
22:11:36 [SENDER] [SEND] seq=4
22:11:36 [SENDER] [ACK] 0
22:11:36 [SENDER] [ACK] 0
22:11:38 [SENDER] [TIMEOUT] Resending window
22:11:38 [SENDER] [RETX] seq=1
22:11:38 [SENDER] [RETX] seq=2
22:11:38 [SENDER] [RETX] seq=3
22:11:38 [SENDER] [LOSS] Dropped packet: DATA-4-dG1lbnQg
22:11:38 [SENDER] [RETX] seq=4
22:11:38 [SENDER] [ACK] 1
22:11:38 [SENDER] [SEND] seq=5
22:11:38 [SENDER] [ACK] 2
22:11:38 [SENDER] [SEND] seq=6
22:11:38 [SENDER] [ACK] 3
22:11:38 [SENDER] [SEND] seq=7
22:11:38 [SENDER] [ACK] 3
22:11:38 [SENDER] [ACK] 3
22:11:38 [SENDER] [ACK] 3
22:11:40 [SENDER] [TIMEOUT] Resending window
22:11:40 [SENDER] [LOSS] Dropped packet: DATA-4-dG1lbnQg
22:11:40 [SENDER] [RETX] seq=4
22:11:40 [SENDER] [RETX] seq=5
22:11:40 [SENDER] [RETX] seq=6
22:11:40 [SENDER] [LOSS] Dropped packet: DATA-7-YW5kIEth
22:11:40 [SENDER] [RETX] seq=7
22:11:40 [SENDER] [ACK] 3
22:11:40 [SENDER] [ACK] 3
22:11:42 [SENDER] [TIMEOUT] Resending window
22:11:42 [SENDER] [RETX] seq=4
22:11:42 [SENDER] [RETX] seq=5
22:11:42 [SENDER] [RETX] seq=6
22:11:42 [SENDER] [LOSS] Dropped packet: DATA-7-YW5kIEth
22:11:42 [SENDER] [RETX] seq=7
22:11:42 [SENDER] [ACK] 4
22:11:42 [SENDER] [SEND] seq=8
22:11:42 [SENDER] [ACK] 5
22:11:42 [SENDER] [SEND] seq=9
22:11:42 [SENDER] [ACK] 6
22:11:42 [SENDER] [SEND] seq=10
22:11:42 [SENDER] [ACK] 6

c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
22:11:36 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
22:11:36 [RECEIVER] [RECV] seq=0, decoded_bytes=512
22:11:36 [RECEIVER] [SEND] ACK-0
22:11:36 [RECEIVER] [RECV] seq=0 (dup)
22:11:36 [RECEIVER] [SEND] ACK-0 (dup)
22:11:38 [RECEIVER] [RECV] seq=1, decoded_bytes=512
22:11:38 [RECEIVER] [SEND] ACK-1
22:11:38 [RECEIVER] [RECV] seq=2, decoded_bytes=512
22:11:38 [RECEIVER] [SEND] ACK-2
22:11:38 [RECEIVER] [RECV] seq=3, decoded_bytes=512
22:11:38 [RECEIVER] [SEND] ACK-3
22:11:38 [RECEIVER] [SEND] ACK-3 (dup)
22:11:38 [RECEIVER] [SEND] ACK-3 (dup)
22:11:38 [RECEIVER] [SEND] ACK-3 (dup)
22:11:40 [RECEIVER] [SEND] ACK-3 (dup)
22:11:40 [RECEIVER] [SEND] ACK-3 (dup)
22:11:42 [RECEIVER] [RECV] seq=4, decoded_bytes=512
22:11:42 [RECEIVER] [SEND] ACK-4
22:11:42 [RECEIVER] [RECV] seq=5, decoded_bytes=512
22:11:42 [RECEIVER] [SEND] ACK-5
22:11:42 [RECEIVER] [RECV] seq=6, decoded_bytes=512
22:11:42 [RECEIVER] [SEND] ACK-6
22:11:42 [RECEIVER] [SEND] ACK-6 (dup)
22:11:42 [RECEIVER] [SEND] ACK-6 (dup)
22:11:42 [RECEIVER] [SEND] ACK-6 (dup)
22:11:44 [RECEIVER] [SEND] ACK-6 (dup)
22:11:44 [RECEIVER] [SEND] ACK-6 (dup)
22:11:44 [RECEIVER] [SEND] ACK-6 (dup)
22:11:46 [RECEIVER] [RECV] seq=7, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-7
22:11:46 [RECEIVER] [RECV] seq=8, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-8
22:11:46 [RECEIVER] [RECV] seq=9, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-9
22:11:46 [RECEIVER] [RECV] seq=10, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-10
22:11:46 [RECEIVER] [RECV] seq=11, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-11
22:11:46 [RECEIVER] [RECV] seq=12, decoded_bytes=512
22:11:46 [RECEIVER] [SEND] ACK-12
22:11:46 [RECEIVER] [RECV] seq=13, decoded_bytes=336
22:11:46 [RECEIVER] [SEND] ACK-13
22:11:47 [RECEIVER] [INFO] Transmission complete (GBN)
22:11:47 [RECEIVER] [INFO] GBN receiver loop ended.

c:\Users\USER\Desktop\ProtocolsProj>
```

The sender window was much larger than the receiver window, so I wasn't able to place them both in the same screen shot, but if the receiver window is sufficient to tell us that packets 1,4,7 were dropped from the transmission.

### 3- DELAY SIMULATION.

For this test we will use a smaller text file.

C:\Windows\system32\cmd.exe	C:\Windows\system32\cmd.exe
22:44:49 [SENDER] [SEND] seq=0	22:44:49 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
22:44:49 [SENDER] [SEND] seq=1	22:44:49 [RECEIVER] [RECV] seq=0, decoded_bytes=512
22:44:49 [SENDER] [SEND] seq=2	22:44:49 [RECEIVER] [SEND] ACK-0
22:44:49 [SENDER] [SEND] seq=3	22:44:49 [RECEIVER] [RECV] seq=1, decoded_bytes=512
22:44:49 [SENDER] [ACK] 0	22:44:49 [RECEIVER] [SEND] ACK-1
22:44:49 [SENDER] [SEND] seq=4	22:44:49 [RECEIVER] [RECV] seq=2, decoded_bytes=512
22:44:49 [SENDER] [ACK] 1	22:44:49 [RECEIVER] [SEND] ACK-2
22:44:49 [SENDER] [SEND] seq=5	22:44:49 [RECEIVER] [RECV] seq=3, decoded_bytes=512
22:44:49 [SENDER] [ACK] 2	22:44:49 [RECEIVER] [SEND] ACK-3
22:44:49 [SENDER] [ACK] 3	22:44:49 [RECEIVER] [RECV] seq=4, decoded_bytes=512
22:44:49 [SENDER] [ACK] 4	22:44:49 [RECEIVER] [SEND] ACK-4
22:44:49 [SENDER] [ACK] 5	22:44:49 [RECEIVER] [RECV] seq=5, decoded_bytes=182
22:44:49 [SENDER] [INFO] Transmission complete (GBN)	22:44:49 [RECEIVER] [SEND] ACK-5
	22:44:49 [RECEIVER] [INFO] Transmission complete (GBN)
c:\Users\USER\Desktop\ProtocolsProj>	22:44:49 [RECEIVER] [INFO] GBN receiver loop ended.
	c:\Users\USER\Desktop\ProtocolsProj>

The test was done with 0.1 seconds TIMEOUT, but it was hard to generate actual late packets, I ran the tests multiple times, but it did not seem like the ACKS were arriving later than 0.1 seconds from the receiver.

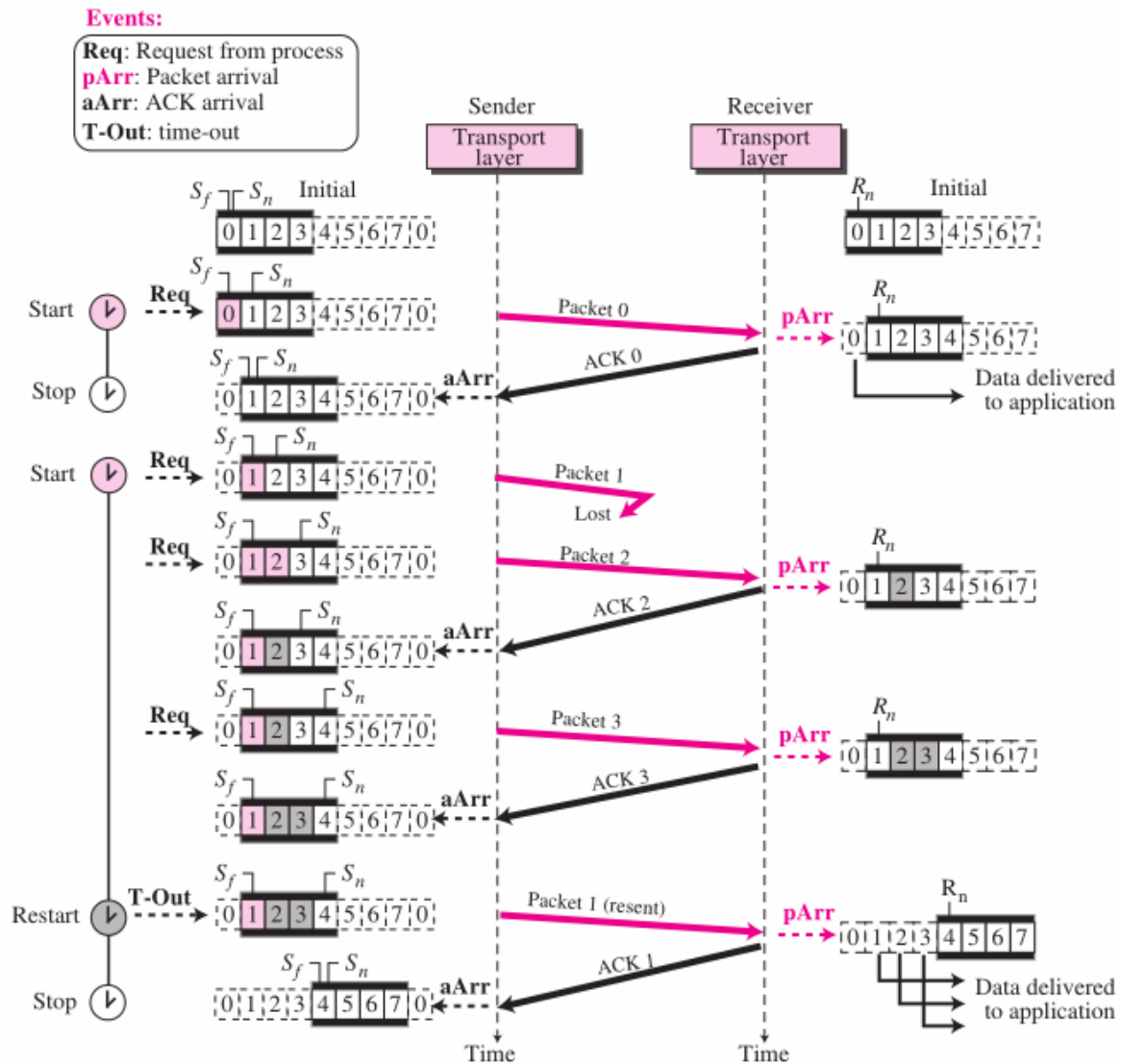
### 4- MIXED CONDITIONS:

Packet loss set to 0.2 and delay to 0.1 seconds

C:\Windows\system32\cmd.exe	C:\Windows\system32\cmd.exe
22:47:45 [SENDER] [SEND] seq=0	22:47:45 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
22:47:45 [SENDER] [LOSS] Dropped packet: DATA-1-YXJkcw0Kb3p	22:47:45 [RECEIVER] [RECV] seq=0, decoded_bytes=512
22:47:45 [SENDER] [SEND] seq=1	22:47:45 [RECEIVER] [SEND] ACK-0
22:47:45 [SENDER] [SEND] seq=2	22:47:45 [RECEIVER] [SEND] ACK-0 (dup)
22:47:45 [SENDER] [SEND] seq=3	22:47:45 [RECEIVER] [SEND] ACK-0 (dup)
22:47:45 [SENDER] [ACK] 0	22:47:45 [RECEIVER] [SEND] ACK-0 (dup)
22:47:45 [SENDER] [SEND] seq=4	22:47:45 [RECEIVER] [RECV] seq=1, decoded_bytes=512
22:47:45 [SENDER] [ACK] 0	22:47:45 [RECEIVER] [SEND] ACK-1
22:47:45 [SENDER] [ACK] 0	22:47:45 [RECEIVER] [RECV] seq=2, decoded_bytes=512
22:47:45 [SENDER] [TIMEOUT] Resending window	22:47:45 [RECEIVER] [SEND] ACK-2
22:47:45 [SENDER] [RETX] seq=1	22:47:45 [RECEIVER] [RECV] seq=3, decoded_bytes=512
22:47:45 [SENDER] [RETX] seq=2	22:47:45 [RECEIVER] [SEND] ACK-3
22:47:45 [SENDER] [RETX] seq=3	22:47:45 [RECEIVER] [RECV] seq=4, decoded_bytes=512
22:47:45 [SENDER] [RETX] seq=4	22:47:45 [RECEIVER] [SEND] ACK-4
22:47:45 [SENDER] [ACK] 1	22:47:45 [RECEIVER] [RECV] seq=5, decoded_bytes=182
22:47:45 [SENDER] [SEND] seq=5	22:47:45 [RECEIVER] [SEND] ACK-5
22:47:45 [SENDER] [ACK] 2	22:47:46 [RECEIVER] [INFO] Transmission complete (GBN)
22:47:45 [SENDER] [ACK] 3	22:47:46 [RECEIVER] [INFO] GBN receiver loop ended.
22:47:45 [SENDER] [ACK] 4	
22:47:45 [SENDER] [ACK] 5	
22:47:46 [SENDER] [INFO] Transmission complete (GBN)	
c:\Users\USER\Desktop\ProtocolsProj>	

The output file was correct and the data was transmitted correctly.

the following diagram from the TCP/IP protocol suite by B.Forouzan shows how the SR protocol works to insure correctness and Out of order delivery of UDP packets to utilize the channel more even if packets were dropped so that we only resend the dropped packets.

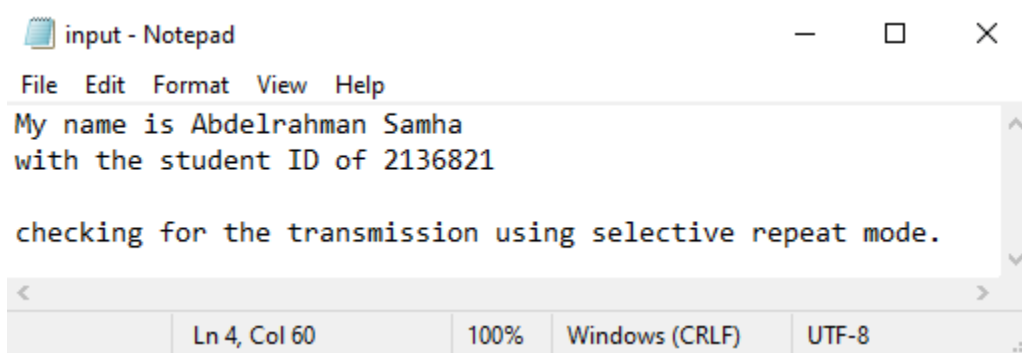




## SR TESTS

### 1 – BASIC TEST

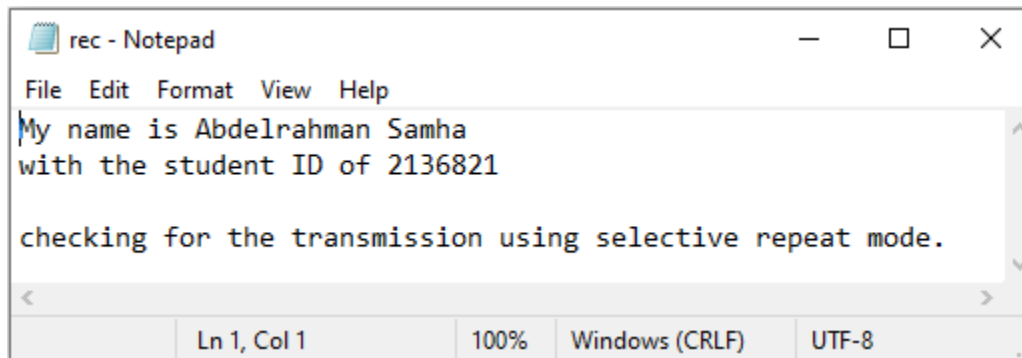
Input :



```
input - Notepad
File Edit Format View Help
My name is Abdelrahman Samha
with the student ID of 2136821

checking for the transmission using selective repeat mode.
Ln 4, Col 60 100% Windows (CRLF) UTF-8
```

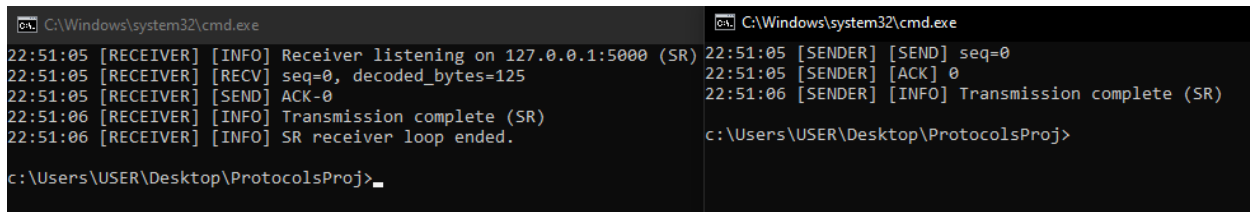
Received :



```
rec - Notepad
File Edit Format View Help
My name is Abdelrahman Samha
with the student ID of 2136821

checking for the transmission using selective repeat mode.
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Transmission :



```
C:\Windows\system32\cmd.exe
22:51:05 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (SR)
22:51:05 [RECEIVER] [RCV] seq=0, decoded_bytes=125
22:51:05 [RECEIVER] [SEND] ACK-0
22:51:06 [RECEIVER] [INFO] Transmission complete (SR)
22:51:06 [RECEIVER] [INFO] SR receiver loop ended.
c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
22:51:05 [SENDER] [SEND] seq=0
22:51:05 [SENDER] [ACK] 0
22:51:06 [SENDER] [INFO] Transmission complete (SR)
c:\Users\USER\Desktop\ProtocolsProj>
```

## 2- LOSS SIMULATION :

Using appricimatly the same intro from the book, the first test is with 0.1 packet loss:

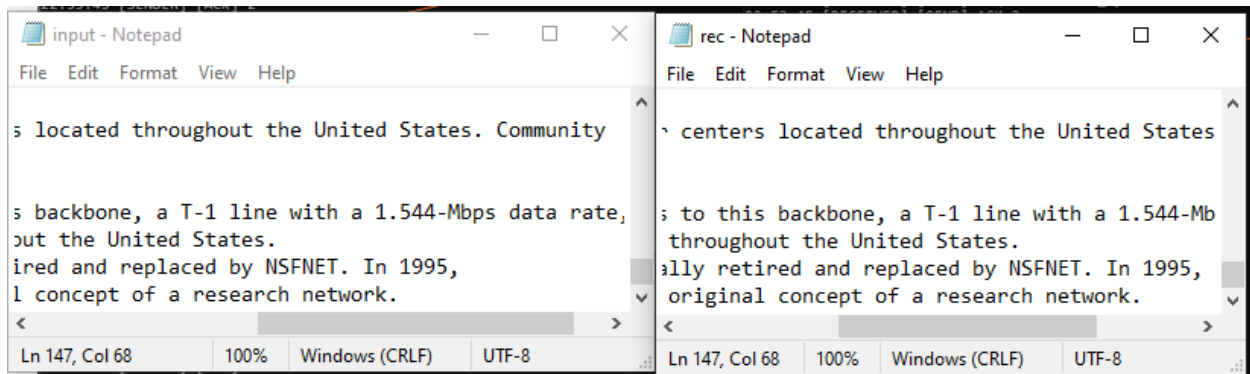
```
C:\Windows\system32\cmd.exe
22:53:45 [SENDER] [SEND] seq=0
22:53:45 [SENDER] [LOSS] Dropped packet: DATA-1-cmRzDQpvcmlnaWShdGVkIGFuZCBtYXR1cm
22:53:45 [SENDER] [SEND] seq=1
22:53:45 [SENDER] [SEND] seq=2
22:53:45 [SENDER] [SEND] seq=3
22:53:45 [SENDER] [ACK] 0
22:53:45 [SENDER] [SEND] seq=4
22:53:45 [SENDER] [ACK] 2
22:53:45 [SENDER] [ACK] 3
22:53:45 [SENDER] [ACK] 4
22:53:47 [SENDER] [RETX] seq=1
22:53:47 [SENDER] [ACK] 1
22:53:47 [SENDER] [SEND] seq=5
22:53:47 [SENDER] [SEND] seq=6
22:53:47 [SENDER] [SEND] seq=7
22:53:47 [SENDER] [SEND] seq=8
22:53:47 [SENDER] [ACK] 5
22:53:47 [SENDER] [SEND] seq=9
22:53:47 [SENDER] [ACK] 6
22:53:47 [SENDER] [SEND] seq=10
22:53:47 [SENDER] [ACK] 7
22:53:47 [SENDER] [LOSS] Dropped packet: DATA-11-QVJQU5FVA0KZm9yIG5vbm1pbG10YXJ5
22:53:47 [SENDER] [SEND] seq=11
22:53:47 [SENDER] [ACK] 8
22:53:47 [SENDER] [SEND] seq=12
22:53:47 [SENDER] [ACK] 9
22:53:47 [SENDER] [SEND] seq=13
22:53:47 [SENDER] [ACK] 10
22:53:47 [SENDER] [ACK] 12
22:53:47 [SENDER] [ACK] 13
22:53:49 [SENDER] [RETX] seq=11
22:53:49 [SENDER] [ACK] 11
22:53:49 [SENDER] [INFO] Transmission complete (SR)

c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
22:53:45 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (SR)
22:53:45 [RECEIVER] [RCV] seq=0, decoded_bytes=512
22:53:45 [RECEIVER] [SEND] ACK-0
22:53:45 [RECEIVER] [RCV] seq=2, decoded_bytes=512
22:53:45 [RECEIVER] [SEND] ACK-2
22:53:45 [RECEIVER] [RCV] seq=3, decoded_bytes=512
22:53:45 [RECEIVER] [SEND] ACK-3
22:53:45 [RECEIVER] [RCV] seq=4, decoded_bytes=512
22:53:45 [RECEIVER] [SEND] ACK-4
22:53:47 [RECEIVER] [RCV] seq=1, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-1
22:53:47 [RECEIVER] [RCV] seq=5, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-5
22:53:47 [RECEIVER] [RCV] seq=6, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-6
22:53:47 [RECEIVER] [RCV] seq=7, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-7
22:53:47 [RECEIVER] [RCV] seq=8, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-8
22:53:47 [RECEIVER] [RCV] seq=9, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-9
22:53:47 [RECEIVER] [RCV] seq=10, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-10
22:53:47 [RECEIVER] [RCV] seq=12, decoded_bytes=512
22:53:47 [RECEIVER] [SEND] ACK-12
22:53:47 [RECEIVER] [RCV] seq=13, decoded_bytes=335
22:53:47 [RECEIVER] [SEND] ACK-13
22:53:49 [RECEIVER] [RCV] seq=11, decoded_bytes=512
22:53:49 [RECEIVER] [SEND] ACK-11
22:53:49 [RECEIVER] [INFO] Transmission complete (SR)
22:53:49 [RECEIVER] [INFO] SR receiver loop ended.

c:\Users\USER\Desktop\ProtocolsProj>
```

Packet 1 was dropped, but was retransmitted later on.



Showing that the data was transmitted correctly.

```
C:\Windows\system32\cmd.exe
22:59:32 [SENDER] [SEND] seq=0
22:59:32 [SENDER] [SEND] seq=1
22:59:32 [SENDER] [SEND] seq=2
22:59:32 [SENDER] [LOSS] Dropped packet: DATA-3-bCBhcyB2YXJpb3ZlIG9yZ2FuaXph
22:59:32 [SENDER] [SEND] seq=3
22:59:32 [SENDER] [ACK] 0
22:59:32 [SENDER] [SEND] seq=4
22:59:32 [SENDER] [ACK] 1
22:59:32 [SENDER] [SEND] seq=5
22:59:32 [SENDER] [ACK] 2
22:59:32 [SENDER] [LOSS] Dropped packet: DATA-6-LQ0kXBVuaNhd6l1vb1BiZXR3ZWU
22:59:32 [SENDER] [SEND] seq=6
22:59:32 [SENDER] [ACK] 4
22:59:32 [SENDER] [ACK] 5
22:59:34 [SENDER] [RETX] seq=3
22:59:34 [SENDER] [RETX] seq=6
22:59:34 [SENDER] [ACK] 3
22:59:34 [SENDER] [SEND] seq=7
22:59:34 [SENDER] [SEND] seq=8
22:59:34 [SENDER] [LOSS] Dropped packet: DATA-9-aGVyZWmdGvYLCBhdXRob3JpdG11
22:59:34 [SENDER] [SEND] seq=9
22:59:34 [SENDER] [ACK] 6
22:59:34 [SENDER] [SEND] seq=10
22:59:34 [SENDER] [ACK] 7
22:59:34 [SENDER] [SEND] seq=11
22:59:34 [SENDER] [ACK] 8
22:59:34 [SENDER] [SEND] seq=12
22:59:34 [SENDER] [ACK] 10
22:59:34 [SENDER] [ACK] 11
22:59:34 [SENDER] [ACK] 12
22:59:36 [SENDER] [RETX] seq=9
22:59:36 [SENDER] [ACK] 9
22:59:36 [SENDER] [SEND] seq=13
22:59:36 [SENDER] [ACK] 13
22:59:36 [SENDER] [INFO] Transmission complete (SR)

C:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
22:59:32 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (SR)
22:59:32 [RECEIVER] [RECV] seq=0, decoded_bytes=512
22:59:32 [RECEIVER] [SEND] ACK-0
22:59:32 [RECEIVER] [RECV] seq=1, decoded_bytes=512
22:59:32 [RECEIVER] [SEND] ACK-1
22:59:32 [RECEIVER] [RECV] seq=2, decoded_bytes=512
22:59:32 [RECEIVER] [SEND] ACK-2
22:59:32 [RECEIVER] [RECV] seq=4, decoded_bytes=512
22:59:32 [RECEIVER] [SEND] ACK-4
22:59:32 [RECEIVER] [RECV] seq=5, decoded_bytes=512
22:59:32 [RECEIVER] [SEND] ACK-5
22:59:34 [RECEIVER] [RECV] seq=3, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-3
22:59:34 [RECEIVER] [RECV] seq=6, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-6
22:59:34 [RECEIVER] [RECV] seq=7, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-7
22:59:34 [RECEIVER] [RECV] seq=8, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-8
22:59:34 [RECEIVER] [RECV] seq=10, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-10
22:59:34 [RECEIVER] [RECV] seq=11, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-11
22:59:34 [RECEIVER] [RECV] seq=12, decoded_bytes=512
22:59:34 [RECEIVER] [SEND] ACK-12
22:59:36 [RECEIVER] [RECV] seq=9, decoded_bytes=512
22:59:36 [RECEIVER] [SEND] ACK-9
22:59:36 [RECEIVER] [RECV] seq=13, decoded_bytes=335
22:59:36 [RECEIVER] [SEND] ACK-13
22:59:36 [RECEIVER] [INFO] Transmission complete (SR)
22:59:36 [RECEIVER] [INFO] SR receiver loop ended.

C:\Users\USER\Desktop\ProtocolsProj>
```

```

C:\Windows\system32\cmd.exe
23:01:27 [SENDER] [SEND] seq=0
23:01:27 [SENDER] [SEND] seq=1
23:01:27 [SENDER] [SEND] seq=2
23:01:27 [SENDER] [SEND] seq=3
23:01:27 [SENDER] [ACK] 0
23:01:27 [SENDER] [SEND] seq=4
23:01:27 [SENDER] [ACK] 1
23:01:27 [SENDER] [SEND] seq=5
23:01:27 [SENDER] [ACK] 2
23:01:27 [SENDER] [SEND] seq=6
23:01:27 [SENDER] [ACK] 3
23:01:27 [SENDER] [SEND] seq=7
23:01:27 [SENDER] [ACK] 4
23:01:27 [SENDER] [SEND] seq=8
23:01:27 [SENDER] [ACK] 5
23:01:27 [SENDER] [SEND] seq=9
23:01:27 [SENDER] [ACK] 6
23:01:27 [SENDER] [SEND] seq=10
23:01:27 [SENDER] [ACK] 7
23:01:27 [SENDER] [SEND] seq=11
23:01:27 [SENDER] [ACK] 8
23:01:27 [SENDER] [SEND] seq=12
23:01:27 [SENDER] [ACK] 9
23:01:27 [SENDER] [SEND] seq=13
23:01:27 [SENDER] [ACK] 10
23:01:27 [SENDER] [ACK] 11
23:01:27 [SENDER] [ACK] 12
23:01:27 [SENDER] [ACK] 13
23:01:28 [SENDER] [INFO] Transmission complete (SR)

c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
23:01:27 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (SR)
23:01:27 [RECEIVER] [RCV] seq=0, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-0
23:01:27 [RECEIVER] [RCV] seq=1, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-1
23:01:27 [RECEIVER] [RCV] seq=2, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-2
23:01:27 [RECEIVER] [RCV] seq=3, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-3
23:01:27 [RECEIVER] [RCV] seq=4, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-4
23:01:27 [RECEIVER] [RCV] seq=5, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-5
23:01:27 [RECEIVER] [RCV] seq=6, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-6
23:01:27 [RECEIVER] [RCV] seq=7, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-7
23:01:27 [RECEIVER] [RCV] seq=8, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-8
23:01:27 [RECEIVER] [RCV] seq=9, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-9
23:01:27 [RECEIVER] [RCV] seq=10, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-10
23:01:27 [RECEIVER] [RCV] seq=11, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-11
23:01:27 [RECEIVER] [RCV] seq=12, decoded_bytes=512
23:01:27 [RECEIVER] [SEND] ACK-12
23:01:27 [RECEIVER] [RCV] seq=13, decoded_bytes=335
23:01:27 [RECEIVER] [SEND] ACK-13
23:01:28 [RECEIVER] [INFO] Transmission complete (SR)
23:01:28 [RECEIVER] [INFO] SR receiver loop ended.

c:\Users\USER\Desktop\ProtocolsProj>

```

Same case happened, it hard to simulate the timeout retransmission without modifying the receiver to delay its ACKs.

#### 4- MIXED CONDITIONS:

Packet loss set to 0.2 and delay to 0.1 seconds

```
C:\Windows\system32\cmd.exe
23:04:32 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (SR)
23:04:32 [RECEIVER] [RECV] seq=0, decoded_bytes=512
23:04:32 [RECEIVER] [SEND] ACK-0
23:04:32 [RECEIVER] [RECV] seq=2, decoded_bytes=512
23:04:32 [RECEIVER] [SEND] ACK-2
23:04:32 [RECEIVER] [RECV] seq=4, decoded_bytes=512
23:04:32 [RECEIVER] [SEND] ACK-4
23:04:33 [RECEIVER] [RECV] seq=1, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-1
23:04:33 [RECEIVER] [RECV] seq=3, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-3
23:04:33 [RECEIVER] [RECV] seq=6, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-6
23:04:33 [RECEIVER] [RECV] seq=7, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-7
23:04:33 [RECEIVER] [RECV] seq=8, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-8
23:04:33 [RECEIVER] [RECV] seq=5, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-5
23:04:33 [RECEIVER] [RECV] seq=9, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-9
23:04:33 [RECEIVER] [RECV] seq=10, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-10
23:04:33 [RECEIVER] [RECV] seq=11, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-11
23:04:33 [RECEIVER] [RECV] seq=12, decoded_bytes=512
23:04:33 [RECEIVER] [SEND] ACK-12
23:04:33 [RECEIVER] [RECV] seq=13, decoded_bytes=335
23:04:33 [RECEIVER] [SEND] ACK-13
23:04:33 [RECEIVER] [INFO] Transmission complete (SR)
23:04:33 [RECEIVER] [INFO] SR receiver loop ended.

c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
23:04:32 [SENDER] [SEND] seq=0
23:04:32 [SENDER] [LOSS] Dropped packet: DATA-1-cmRzDQpvcmlnaW5hdGvKIGFuZCBtYXR1cmVkJiBBbHN...
23:04:32 [SENDER] [SEND] seq=1
23:04:32 [SENDER] [SEND] seq=2
23:04:32 [SENDER] [LOSS] Dropped packet: DATA-3-bCBhcyB2YXJpb3VzIG9yZ2FuaXphdGlvbnMgc3VjaCB...
23:04:32 [SENDER] [SEND] seq=3
23:04:32 [SENDER] [ACK] 0
23:04:32 [SENDER] [SEND] seq=4
23:04:32 [SENDER] [ACK] 2
23:04:32 [SENDER] [ACK] 4
23:04:32 [SENDER] [LOSS] Dropped packet: DATA-1-cmRzDQpvcmlnaW5hdGvKIGFuZCBtYXR1cmVkJiBBbHN...
23:04:32 [SENDER] [RETX] seq=1
23:04:32 [SENDER] [LOSS] Dropped packet: DATA-3-bCBhcyB2YXJpb3VzIG9yZ2FuaXphdGlvbnMgc3VjaCB...
23:04:32 [SENDER] [RETX] seq=3
23:04:33 [SENDER] [RETX] seq=1
23:04:33 [SENDER] [RETX] seq=3
23:04:33 [SENDER] [ACK] 1
23:04:33 [SENDER] [LOSS] Dropped packet: DATA-5-Y2UgbWVzc2FnZSBwcm9jZHNzb3INCihJTApLiBUaGU...
23:04:33 [SENDER] [SEND] seq=5
23:04:33 [SENDER] [SEND] seq=6
23:04:33 [SENDER] [ACK] 3
23:04:33 [SENDER] [SEND] seq=7
23:04:33 [SENDER] [SEND] seq=8
23:04:33 [SENDER] [ACK] 6
23:04:33 [SENDER] [ACK] 7
23:04:33 [SENDER] [ACK] 8
23:04:33 [SENDER] [RETX] seq=5
23:04:33 [SENDER] [ACK] 5
23:04:33 [SENDER] [SEND] seq=9
23:04:33 [SENDER] [SEND] seq=10
23:04:33 [SENDER] [SEND] seq=11
23:04:33 [SENDER] [SEND] seq=12
23:04:33 [SENDER] [ACK] 9
23:04:33 [SENDER] [SEND] seq=13
23:04:33 [SENDER] [ACK] 10
23:04:33 [SENDER] [ACK] 11
23:04:33 [SENDER] [ACK] 12
23:04:33 [SENDER] [ACK] 13
23:04:33 [SENDER] [INFO] Transmission complete (SR)

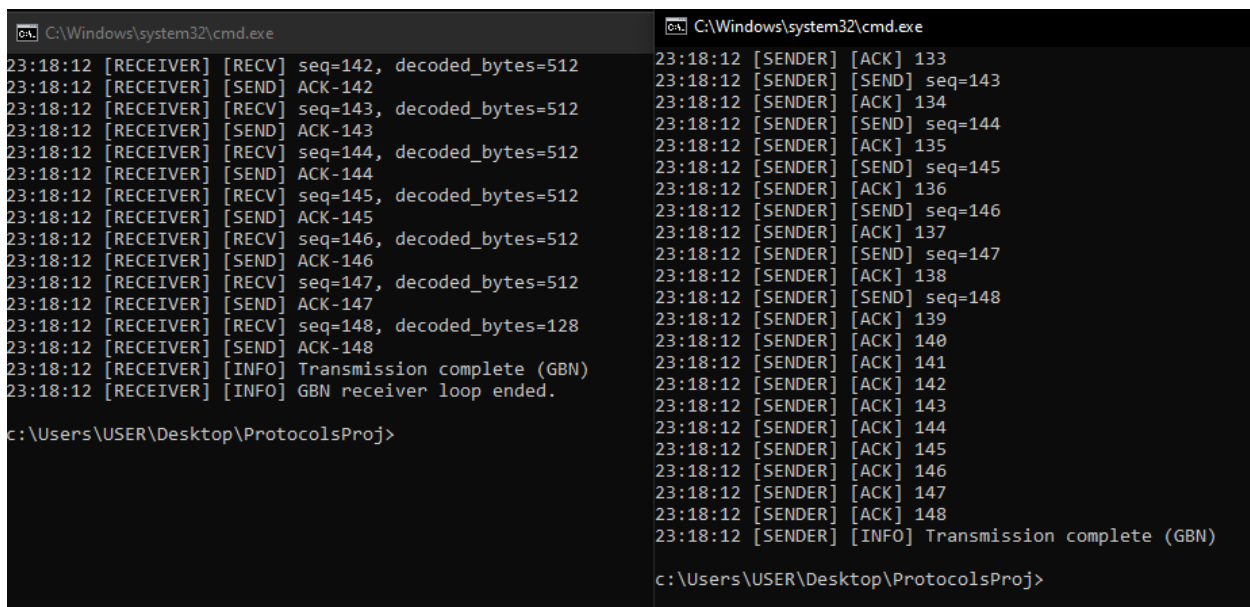
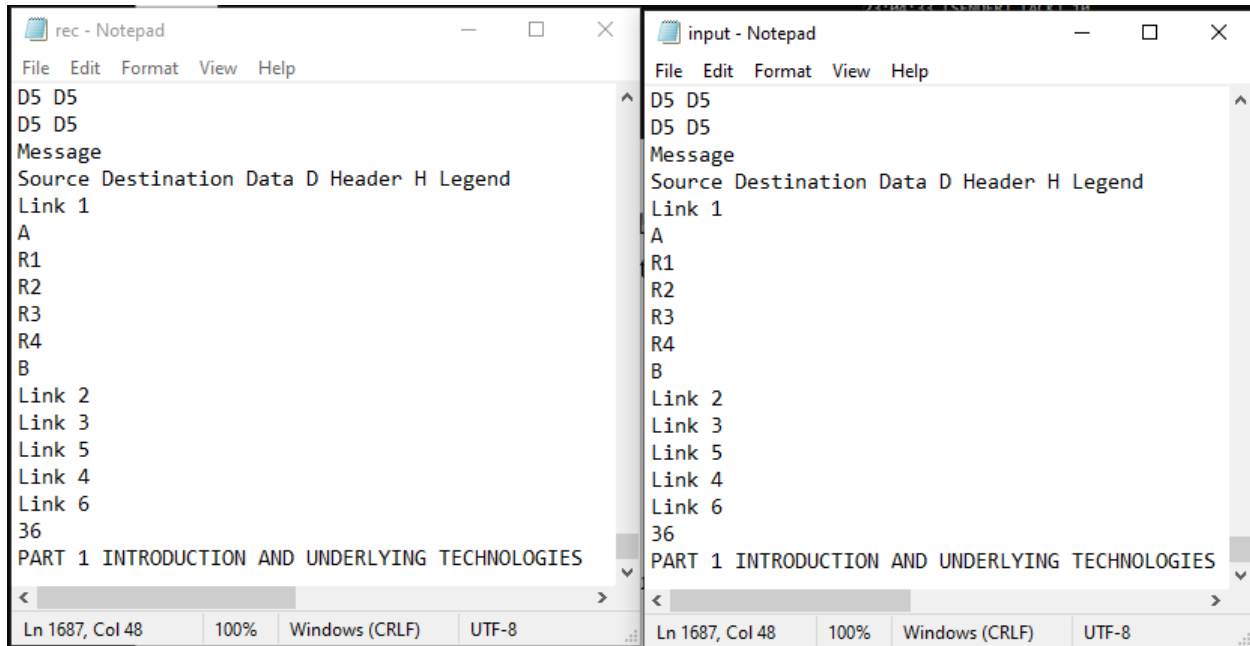
c:\Users\USER\Desktop\ProtocolsProj>
```

A great case happened here, were packet 1 was dropped twice in a row, but was still retransmitted, this shows how robust the implementation is.

## Stress test.

For this test, we are going to get a larger text from the same book used before, to test the system heavily.

For GBN I copied the first 2 chapter from the TCP/IP book and here is the result using window size of 10, and no packets are lost:



FOR SR using the same conditions and input file

```
input - Notepad
File Edit Format View Help
D5 D5
D5 D5
Message
Source Destination Data D Header H Legend
Link 1
A
R1
R2
R3
R4
B
Link 2
Link 3
Link 5
Link 4
Link 6
36
PART 1 INTRODUCTION AND UNDERLYING TECHNOLOGIES
Ln 1687, Col 48 100% Windows (CRLF) UTF-8

rec - Notepad
File Edit Format View Help
5 D5
5 D5
Message
Source Destination Data D Header H Legend
Link 1
1
2
3
4
Link 2
Link 3
Link 5
Link 4
Link 6
6
PART 1 INTRODUCTION AND UNDERLYING TECHNOLOGIES
Ln 1687, Col 48 100% Windows (CRLF) UTF-8
```

```
C:\Windows\system32\cmd.exe
23:20:56 [RECEIVER] [SEND] ACK-137
23:20:56 [RECEIVER] [RCV] seq=138, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-138
23:20:56 [RECEIVER] [RCV] seq=139, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-139
23:20:56 [RECEIVER] [RCV] seq=140, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-140
23:20:56 [RECEIVER] [RCV] seq=141, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-141
23:20:56 [RECEIVER] [RCV] seq=142, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-142
23:20:56 [RECEIVER] [RCV] seq=143, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-143
23:20:56 [RECEIVER] [RCV] seq=144, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-144
23:20:56 [RECEIVER] [RCV] seq=145, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-145
23:20:56 [RECEIVER] [RCV] seq=146, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-146
23:20:56 [RECEIVER] [RCV] seq=147, decoded_bytes=512
23:20:56 [RECEIVER] [SEND] ACK-147
23:20:56 [RECEIVER] [RCV] seq=148, decoded_bytes=128
23:20:56 [RECEIVER] [SEND] ACK-148
23:20:57 [RECEIVER] [INFO] Transmission complete (SR)
23:20:57 [RECEIVER] [INFO] SR receiver loop ended.
c:\Users\USER\Desktop\ProtocolsProj>

C:\Windows\system32\cmd.exe
23:20:56 [SENDER] [ACK] 131
23:20:56 [SENDER] [SEND] seq=141
23:20:56 [SENDER] [ACK] 132
23:20:56 [SENDER] [SEND] seq=142
23:20:56 [SENDER] [ACK] 133
23:20:56 [SENDER] [SEND] seq=143
23:20:56 [SENDER] [ACK] 134
23:20:56 [SENDER] [SEND] seq=144
23:20:56 [SENDER] [ACK] 135
23:20:56 [SENDER] [SEND] seq=145
23:20:56 [SENDER] [ACK] 136
23:20:56 [SENDER] [SEND] seq=146
23:20:56 [SENDER] [ACK] 137
23:20:56 [SENDER] [SEND] seq=147
23:20:56 [SENDER] [ACK] 138
23:20:56 [SENDER] [SEND] seq=148
23:20:56 [SENDER] [ACK] 139
23:20:56 [SENDER] [ACK] 140
23:20:56 [SENDER] [ACK] 141
23:20:56 [SENDER] [ACK] 142
23:20:56 [SENDER] [ACK] 143
23:20:56 [SENDER] [ACK] 144
23:20:56 [SENDER] [ACK] 145
23:20:56 [SENDER] [ACK] 146
23:20:56 [SENDER] [ACK] 147
23:20:56 [SENDER] [ACK] 148
23:20:57 [SENDER] [INFO] Transmission complete (SR)
c:\Users\USER\Desktop\ProtocolsProj>
```

# How to operate

Using the CLI do the same commands as follows :

```
Command Prompt
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER>cd desktop

C:\Users\USER\Desktop>cd protocolsproj

C:\Users\USER\Desktop\ProtocolsProj>python executable.py
Enter mode (gbn/sr): gbn
Enter window size: 10
Enter packet loss %: 0.1
Enter delay : 2

C:\Users\USER\Desktop\ProtocolsProj>
```

The executable.py is a script that opens two terminals and it start the receiver program first then the sender program with the same configuration inputs from the user and runs the data transmission protocols. This program should get the path of where the folder of the programs are stored, if a problem occurred during that, the user can open 2 separate terminals and manually enter the inputs as follows, starting with the receiver :

```
C:\Windows\system32\cmd.exe

C:\Users\USER\Desktop\ProtocolsProj>python receiver.py
Usage: python receiver.py <port> <mode: gbn/sr> <outfile>

C:\Users\USER\Desktop\ProtocolsProj>python receiver.py 5000 gbn rec.txt
23:41:41 [RECEIVER] [INFO] Receiver listening on 127.0.0.1:5000 (GBN)
```

Entering wrong inputs will make the program show what is required, you must launch the receiver program first before the sender

The sender terminal :

```
C:\Windows\system32\cmd.exe

C:\Users\USER\Desktop\ProtocolsProj>python sender.py
Usage: python sender.py <receiver_ip> <receiver_port> <mode: gbn/sr> <window_size> <loss_rate> <timeout> <filename>

C:\Users\USER\Desktop\ProtocolsProj>python sender.py 127.0.0.1 5000 gbn 10 0 2 input.txt
23:42:05 [SENDER] [SEND] seq=0
23:42:05 [SENDER] [SEND] seq=1
23:42:05 [SENDER] [SEND] seq=2
```