Go-Back-N and Selective Repeat Protocols

Programming Language

Python Version 2.7
(Note:** The code does not work in Python 3.5**)

Objective

The objective of this project is to implement sliding window protocols – Go-Back-N and Selective Repeat protocols using an Unreliable Datagram Protocol (UDP) method. Our goal in project is to simulate checksum error, packet loss and ACK loss during the data transfer between sender and receiver.

Unreliable Datagram Protocol (UDP)

UDP is a transport layer protocol that provides best effort is transferring data between systems. UDP offers unreliable service that provides no guarantee in data delivery and prevention of data duplication. Compared to other reliable data transfer methods like TCP, UDP has unique way of data transfer. UDP does not establish connection between End-to-End systems and does not provide any form of congestion control or communication security for the data transfer. These characteristics contributes to efficient data communication for some application like media streaming and online gaming.
We used existing socket programming from Project-1 and UDP connection on top of that to achieve our goal in this project.

Difference between Go-Back-N and Selective Repeat Protocols

	Go-Back-N Protocol	Selective Repeat Protocol
Functionality	Go-Back-N protocol retransmits all frames that are sent after a frame that was suspected to be lost or corrupted	Selective Repeat protocol retransmit only the frames that are suspected to be lost or corrupted
Sender Window Size	Maximum value of N-1, where N is number of sequence numbers	Maximum value of (N/2), where N is number of sequence numbers
Receiver Window Size	1	Maximum value of (N/2), where N is number of sequence numbers
Receiver Storage	Since receiver's maximum window size is 1, it does not require any storage for received frames	Receiver stores all frames that are received after receiving a corrupted frame until the corrupted frame is replaced by a valid frame

Bandwidth Utilization	Bandwidth utilization is high if the error rate is high since the protocol need to send all frames even though received correctly after the reception of corrupted frame	Bandwidth utilization is minimum since the protocol retransmits only corrupted frames

Reference: ***http://techdifferences.com/difference-between-go-back-n-and-selective-repeat-protocol.html***

Implementation and Execution

As mentioned in the project requirements, following probability setup is made in both protocols

- **Probability of checksum error** is set to **0.1**
- **Probability of packet loss** is set to **0.1**
- **Probability of ACK loss** is set to **0.05**
- Checksum error is introduced by the sender before sending the packet to the receiver. Receiver detects problem in the packet and requests the same packet
- Similarly, ACK loss is handled in the Sender side. When ACK is received, sender purposefully drops the ACK and waits for timeout of the packet. Once the timeout occurs, sender resends the packet
- Packet loss is handled in the Receiver side. When a packet is received, receiver purposefully drops the packet. Sender waits for ACK for the sent packet until timeout. Once the timeout occurs, sender resends the packet

1. Go-Back-N

- Start Receiver with the command:
  ```
  python ServerApp.py -a <Sender-IP> -b <Sender-Port> -x <Receiver-IP> -y <Receiver-Port> -m <Max_Sequence_Number_Bits> -t <Timeout>
  ```

- Start Sender with the command:
  ```
  python ClientApp.py -f <FileName>-a <Sender-IP> -b <Sender-Port> -x <Receiver-IP> -y <Receiver-Port> -m <Max_Sequence_Number_Bits>  -s <Max_Segment_Size> -n <Total_Packets> -t <Timeout>
  ```
  where **Max_Sequence_Number_Bits** is number of sequence numbers to be considered, **Max_Segment_Size** is number of segments the given file have to be split and **Total_Packets** is total packets of the file to be sent

- All the above values are set to default values. Default values are:
  - Sender-IP: 127.0.0.1
  - Receiver-IP: 127.0.0.1
  - Sender-Port: 8081
  - Receiver-Port: 8080
  - Max_Sequence_Number_Bits: 2

- Max_Segment_Size: 1500
- Total_Packets: ALL
- Timeout: 10
- File_Name: Index.html (Attached in the project)

  Note: The project also contains default file locations (Data folder in the project) for the sender and receiver.

- Simple execution commands:

  Receiver:

  python ServerApp.py -a 127.0.0.1 -b 8081 -x 127.0.0.1 -y 8080 -m 3

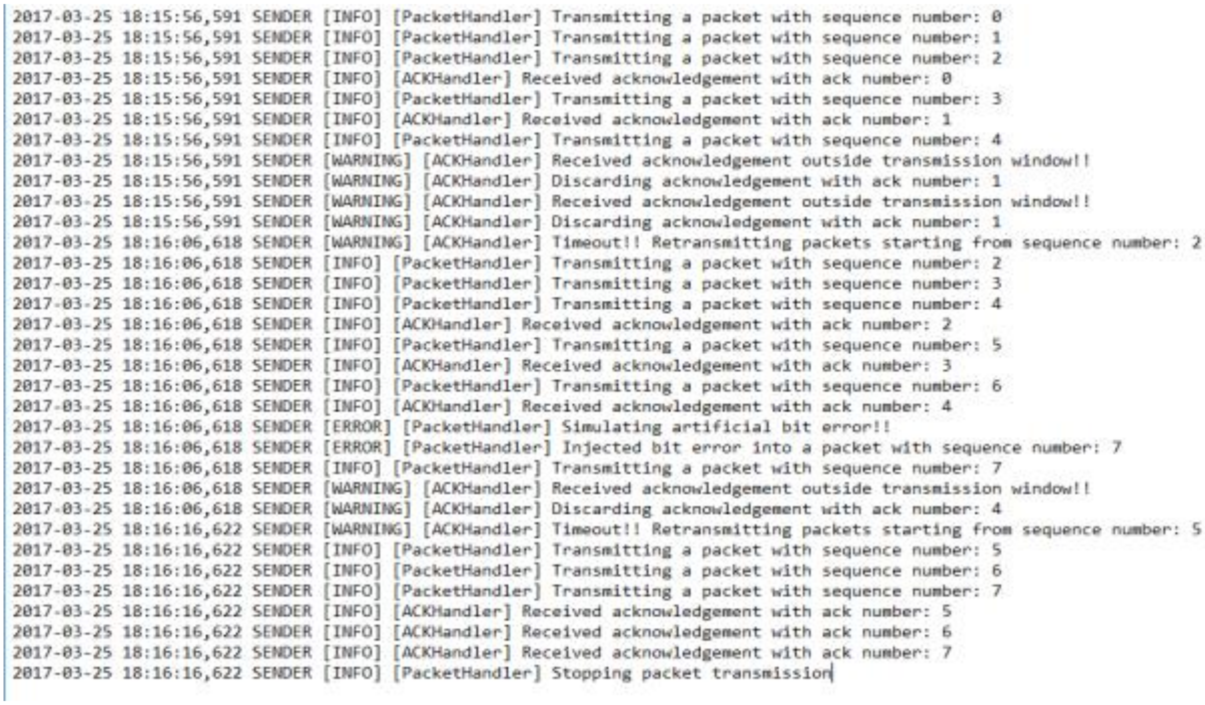
  Sender:

  python ClientApp.py -f index.html -a 127.0.0.1 -b 8081 -x 127.0.0.1 -y 8080 -m 3 -s 1600 -t 15

  (***Note:*** *-m <Max_Sequence_Number_Bits> has to be set as equal values in both sender and receiver end*)

- Results for above commands are given in Figure-1 for Sender and Figure-2 for Receiver:

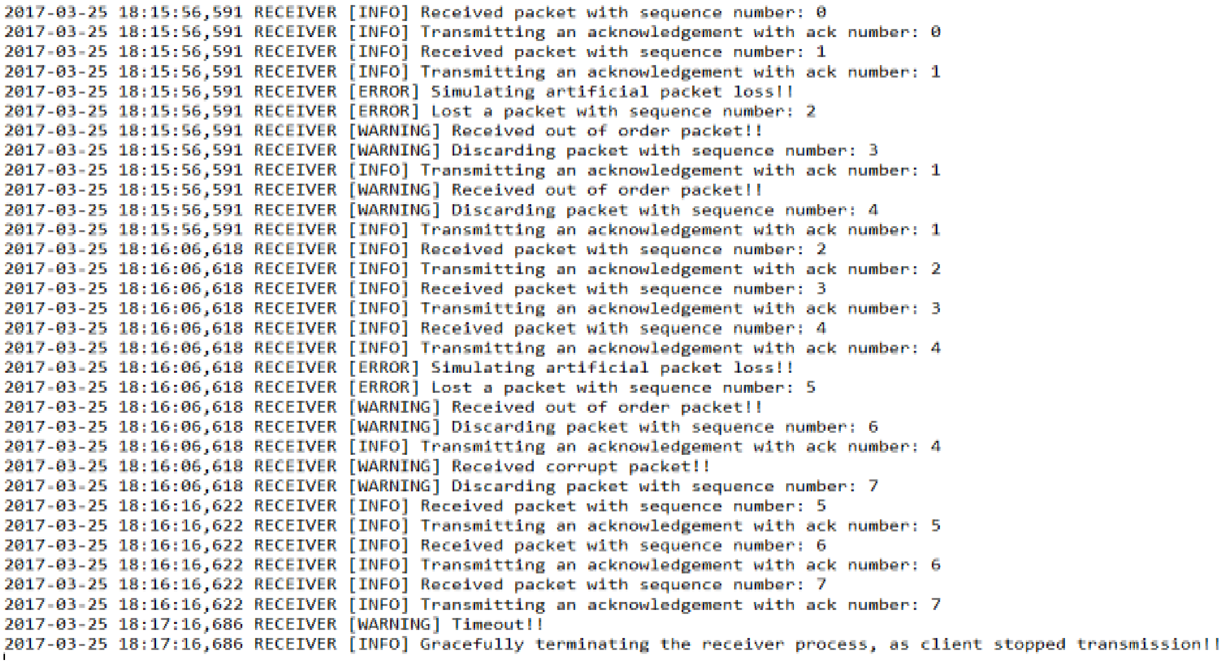
  Sender:



```
2017-03-25 18:15:56,591 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 0
2017-03-25 18:15:56,591 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 1
2017-03-25 18:15:56,591 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 2
2017-03-25 18:15:56,591 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 0
2017-03-25 18:15:56,591 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 3
2017-03-25 18:15:56,591 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 1
2017-03-25 18:15:56,591 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 4
2017-03-25 18:15:56,591 SENDER [WARNING] [ACKHandler] Received acknowledgement outside transmission window!!
2017-03-25 18:15:56,591 SENDER [WARNING] [ACKHandler] Discarding acknowledgement with ack number: 1
2017-03-25 18:15:56,591 SENDER [WARNING] [ACKHandler] Received acknowledgement outside transmission window!!
2017-03-25 18:15:56,591 SENDER [WARNING] [ACKHandler] Discarding acknowledgement with ack number: 1
2017-03-25 18:16:06,618 SENDER [WARNING] [ACKHandler] Timeout!! Retransmitting packets starting from sequence number: 2
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 2
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 3
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 4
2017-03-25 18:16:06,618 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 2
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 5
2017-03-25 18:16:06,618 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 3
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 6
2017-03-25 18:16:06,618 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 4
2017-03-25 18:16:06,618 SENDER [ERROR] [PacketHandler] Simulating artificial bit error!!
2017-03-25 18:16:06,618 SENDER [ERROR] [PacketHandler] Injected bit error into a packet with sequence number: 7
2017-03-25 18:16:06,618 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 7
2017-03-25 18:16:06,618 SENDER [WARNING] [ACKHandler] Received acknowledgement outside transmission window!!
2017-03-25 18:16:06,618 SENDER [WARNING] [ACKHandler] Discarding acknowledgement with ack number: 4
2017-03-25 18:16:16,622 SENDER [WARNING] [ACKHandler] Timeout!! Retransmitting packets starting from sequence number: 5
2017-03-25 18:16:16,622 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 5
2017-03-25 18:16:16,622 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 6
2017-03-25 18:16:16,622 SENDER [INFO] [PacketHandler] Transmitting a packet with sequence number: 7
2017-03-25 18:16:16,622 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 5
2017-03-25 18:16:16,622 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 6
2017-03-25 18:16:16,622 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 7
2017-03-25 18:16:16,622 SENDER [INFO] [PacketHandler] Stopping packet transmission
```

Figure 1: Go-Back-N Client Execution for the given command with bit error and ACK loss simulation

- **Receiver:**



```
2017-03-25 18:15:56,591 RECEIVER [INFO] Received packet with sequence number: 0
2017-03-25 18:15:56,591 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 0
2017-03-25 18:15:56,591 RECEIVER [INFO] Received packet with sequence number: 1
2017-03-25 18:15:56,591 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 1
2017-03-25 18:15:56,591 RECEIVER [ERROR] Simulating artificial packet loss!!
2017-03-25 18:15:56,591 RECEIVER [ERROR] Lost a packet with sequence number: 2
2017-03-25 18:15:56,591 RECEIVER [WARNING] Received out of order packet!!
2017-03-25 18:15:56,591 RECEIVER [WARNING] Discarding packet with sequence number: 3
2017-03-25 18:15:56,591 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 1
2017-03-25 18:15:56,591 RECEIVER [WARNING] Received out of order packet!!
2017-03-25 18:15:56,591 RECEIVER [WARNING] Discarding packet with sequence number: 4
2017-03-25 18:15:56,591 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 1
2017-03-25 18:16:06,618 RECEIVER [INFO] Received packet with sequence number: 2
2017-03-25 18:16:06,618 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 2
2017-03-25 18:16:06,618 RECEIVER [INFO] Received packet with sequence number: 3
2017-03-25 18:16:06,618 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 3
2017-03-25 18:16:06,618 RECEIVER [INFO] Received packet with sequence number: 4
2017-03-25 18:16:06,618 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 4
2017-03-25 18:16:06,618 RECEIVER [ERROR] Simulating artificial packet loss!!
2017-03-25 18:16:06,618 RECEIVER [ERROR] Lost a packet with sequence number: 5
2017-03-25 18:16:06,618 RECEIVER [WARNING] Received out of order packet!!
2017-03-25 18:16:06,618 RECEIVER [WARNING] Discarding packet with sequence number: 6
2017-03-25 18:16:06,618 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 4
2017-03-25 18:16:06,618 RECEIVER [WARNING] Received corrupt packet!!
2017-03-25 18:16:06,618 RECEIVER [WARNING] Discarding packet with sequence number: 7
2017-03-25 18:16:16,622 RECEIVER [INFO] Received packet with sequence number: 5
2017-03-25 18:16:16,622 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 5
2017-03-25 18:16:16,622 RECEIVER [INFO] Received packet with sequence number: 6
2017-03-25 18:16:16,622 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 6
2017-03-25 18:16:16,622 RECEIVER [INFO] Received packet with sequence number: 7
2017-03-25 18:16:16,622 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 7
2017-03-25 18:17:16,686 RECEIVER [WARNING] Timeout!!
2017-03-25 18:17:16,686 RECEIVER [INFO] Gracefully terminating the receiver process, as client stopped transmission!!
```

Figure 2: Go-Back-N Receiver side execution for the given command with packet loss simulation

Note: Once the Receiver terminates, the transferred file will be available in Receiver's location.

2. Selective Repeat

- Start Receiver with the command:
  python ServerApp.py -a <Sender-IP> -b <Sender-Port> -x <Receiver-IP> -y <Receiver-Port> -m <Max_Sequence_Number_Bits> -t <Timeout>

- Start Sender with the command:
  python ClientApp.py -f <FileName>-a <Sender-IP> -b <Sender-Port> -x <Receiver-IP> -y <Receiver-Port> -m <Max_Sequence_Number_Bits> -s <Max_Segment_Size> -n <Total_Packets> -t <Timeout>

  where **Max_Sequence_Number_Bits** is number of sequence numbers to be considered, **Max_Segment_Size** is number of segments the given file have to be split and **Total_Packets** is total packets of the file to be sent

- All the above values are set to default values. Default values are:
  - Sender-IP: 127.0.0.1
  - Receiver-IP: 127.0.0.1
  - Sender-Port: 8081
  - Receiver-Port: 8080
  - Max_Sequence_Number_Bits: 2
  - Max_Segment_Size: 1500
  - Total_Packets: ALL

- Timeout: 10
- File_Name: Index.html (Attached in the project)

  Note: The project also contains default file locations (Data folder in the project) for the sender and receiver.

- Simple execution commands:

  Receiver:

  python ServerApp.py -a 127.0.0.1 -b 8081 -x 127.0.0.1 -y 8080 -m 3

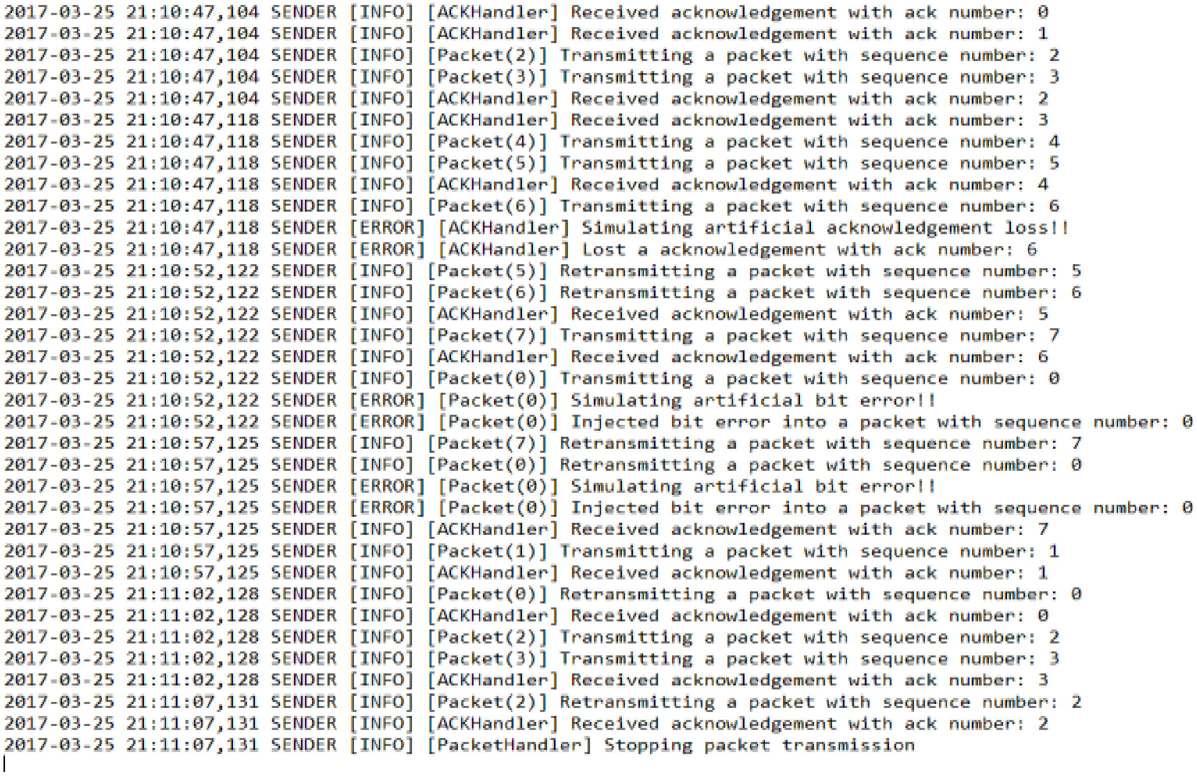
  Sender:

  python ClientApp.py -f index.html -a 127.0.0.1 -b 8081 -x 127.0.0.1 -y 8080 -m 3 -s 1000 -n 20 -t 5

  (***Note:*** *-m <Max_Sequence_Number_Bits> has to be set as equal values in both sender and receiver end*)

- Results for above commands are given in Figure-3 for Sender and Figure-4 for Receiver:

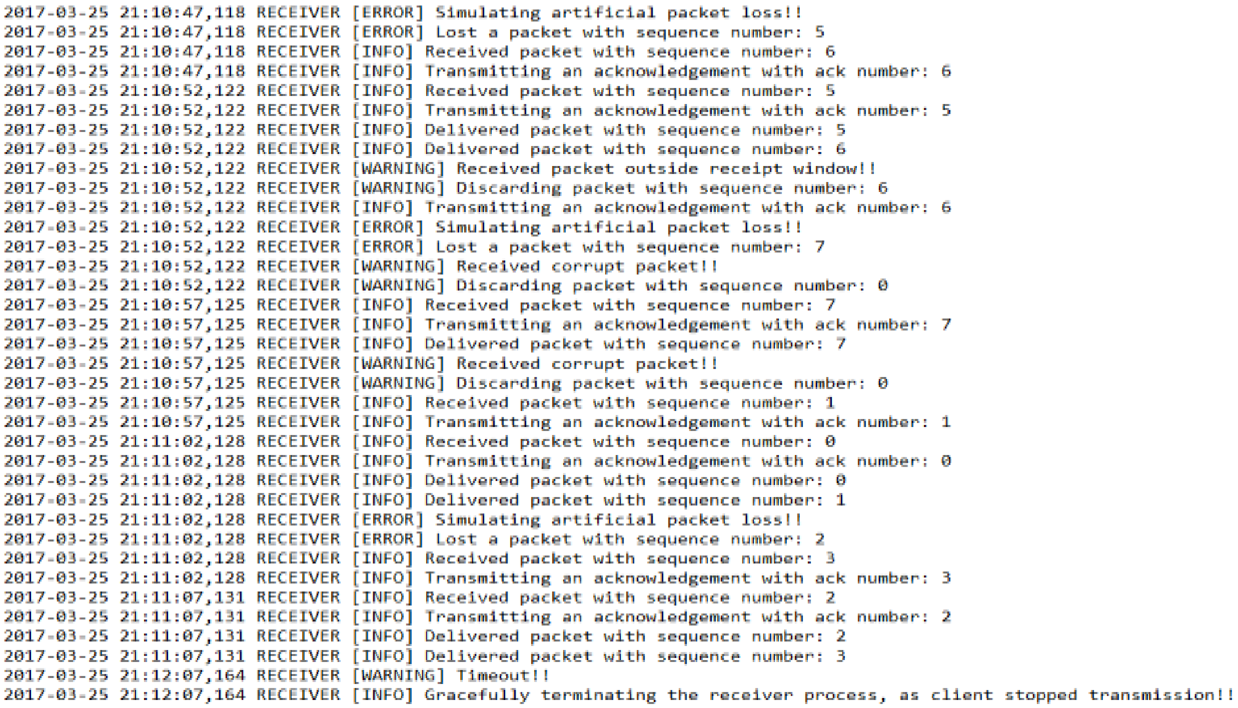
  Sender:



```
2017-03-25 21:10:47,104 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 0
2017-03-25 21:10:47,104 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 1
2017-03-25 21:10:47,104 SENDER [INFO] [Packet(2)] Transmitting a packet with sequence number: 2
2017-03-25 21:10:47,104 SENDER [INFO] [Packet(3)] Transmitting a packet with sequence number: 3
2017-03-25 21:10:47,104 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 2
2017-03-25 21:10:47,118 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 3
2017-03-25 21:10:47,118 SENDER [INFO] [Packet(4)] Transmitting a packet with sequence number: 4
2017-03-25 21:10:47,118 SENDER [INFO] [Packet(5)] Transmitting a packet with sequence number: 5
2017-03-25 21:10:47,118 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 4
2017-03-25 21:10:47,118 SENDER [INFO] [Packet(6)] Transmitting a packet with sequence number: 6
2017-03-25 21:10:47,118 SENDER [ERROR] [ACKHandler] Simulating artificial acknowledgement loss!!
2017-03-25 21:10:47,118 SENDER [ERROR] [ACKHandler] Lost a acknowledgement with ack number: 6
2017-03-25 21:10:52,122 SENDER [INFO] [Packet(5)] Retransmitting a packet with sequence number: 5
2017-03-25 21:10:52,122 SENDER [INFO] [Packet(6)] Retransmitting a packet with sequence number: 6
2017-03-25 21:10:52,122 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 5
2017-03-25 21:10:52,122 SENDER [INFO] [Packet(7)] Transmitting a packet with sequence number: 7
2017-03-25 21:10:52,122 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 6
2017-03-25 21:10:52,122 SENDER [INFO] [Packet(0)] Transmitting a packet with sequence number: 0
2017-03-25 21:10:52,122 SENDER [ERROR] [Packet(0)] Simulating artificial bit error!!
2017-03-25 21:10:52,122 SENDER [ERROR] [Packet(0)] Injected bit error into a packet with sequence number: 0
2017-03-25 21:10:57,125 SENDER [INFO] [Packet(7)] Retransmitting a packet with sequence number: 7
2017-03-25 21:10:57,125 SENDER [INFO] [Packet(0)] Retransmitting a packet with sequence number: 0
2017-03-25 21:10:57,125 SENDER [ERROR] [Packet(0)] Simulating artificial bit error!!
2017-03-25 21:10:57,125 SENDER [ERROR] [Packet(0)] Injected bit error into a packet with sequence number: 0
2017-03-25 21:10:57,125 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 7
2017-03-25 21:10:57,125 SENDER [INFO] [Packet(1)] Transmitting a packet with sequence number: 1
2017-03-25 21:10:57,125 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 1
2017-03-25 21:11:02,128 SENDER [INFO] [Packet(0)] Retransmitting a packet with sequence number: 0
2017-03-25 21:11:02,128 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 0
2017-03-25 21:11:02,128 SENDER [INFO] [Packet(2)] Transmitting a packet with sequence number: 2
2017-03-25 21:11:02,128 SENDER [INFO] [Packet(3)] Transmitting a packet with sequence number: 3
2017-03-25 21:11:02,128 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 3
2017-03-25 21:11:07,131 SENDER [INFO] [Packet(2)] Retransmitting a packet with sequence number: 2
2017-03-25 21:11:07,131 SENDER [INFO] [ACKHandler] Received acknowledgement with ack number: 2
2017-03-25 21:11:07,131 SENDER [INFO] [PacketHandler] Stopping packet transmission
```

Figure 3 Selective Repeat Sender side execution with ACK loss and bit error simulation

Receiver:



```
2017-03-25 21:10:47,118 RECEIVER [ERROR] Simulating artificial packet loss!!
2017-03-25 21:10:47,118 RECEIVER [ERROR] Lost a packet with sequence number: 5
2017-03-25 21:10:47,118 RECEIVER [INFO] Received packet with sequence number: 6
2017-03-25 21:10:47,118 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 6
2017-03-25 21:10:52,122 RECEIVER [INFO] Received packet with sequence number: 5
2017-03-25 21:10:52,122 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 5
2017-03-25 21:10:52,122 RECEIVER [INFO] Delivered packet with sequence number: 5
2017-03-25 21:10:52,122 RECEIVER [INFO] Delivered packet with sequence number: 6
2017-03-25 21:10:52,122 RECEIVER [WARNING] Received packet outside receipt window!!
2017-03-25 21:10:52,122 RECEIVER [WARNING] Discarding packet with sequence number: 6
2017-03-25 21:10:52,122 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 6
2017-03-25 21:10:52,122 RECEIVER [ERROR] Simulating artificial packet loss!!
2017-03-25 21:10:52,122 RECEIVER [ERROR] Lost a packet with sequence number: 7
2017-03-25 21:10:52,122 RECEIVER [WARNING] Received corrupt packet!!
2017-03-25 21:10:52,122 RECEIVER [WARNING] Discarding packet with sequence number: 0
2017-03-25 21:10:57,125 RECEIVER [INFO] Received packet with sequence number: 7
2017-03-25 21:10:57,125 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 7
2017-03-25 21:10:57,125 RECEIVER [INFO] Delivered packet with sequence number: 7
2017-03-25 21:10:57,125 RECEIVER [WARNING] Received corrupt packet!!
2017-03-25 21:10:57,125 RECEIVER [WARNING] Discarding packet with sequence number: 0
2017-03-25 21:10:57,125 RECEIVER [INFO] Received packet with sequence number: 1
2017-03-25 21:10:57,125 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 1
2017-03-25 21:11:02,128 RECEIVER [INFO] Received packet with sequence number: 0
2017-03-25 21:11:02,128 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 0
2017-03-25 21:11:02,128 RECEIVER [INFO] Delivered packet with sequence number: 0
2017-03-25 21:11:02,128 RECEIVER [INFO] Delivered packet with sequence number: 1
2017-03-25 21:11:02,128 RECEIVER [ERROR] Simulating artificial packet loss!!
2017-03-25 21:11:02,128 RECEIVER [ERROR] Lost a packet with sequence number: 2
2017-03-25 21:11:02,128 RECEIVER [INFO] Received packet with sequence number: 3
2017-03-25 21:11:02,128 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 3
2017-03-25 21:11:07,131 RECEIVER [INFO] Received packet with sequence number: 2
2017-03-25 21:11:07,131 RECEIVER [INFO] Transmitting an acknowledgement with ack number: 2
2017-03-25 21:11:07,131 RECEIVER [INFO] Delivered packet with sequence number: 2
2017-03-25 21:11:07,131 RECEIVER [INFO] Delivered packet with sequence number: 3
2017-03-25 21:12:07,164 RECEIVER [WARNING] Timeout!!
2017-03-25 21:12:07,164 RECEIVER [INFO] Gracefully terminating the receiver process, as client stopped transmission!!
```

Figure 4: Selective Repeat Receiver side execution for the given command with Packet loss simulation

Note: Once the Receiver terminates, the transferred file will be available in Receiver’s location.