Name: Rupesh Hemant Bhusare, roll no.: 2203106

LAB 5: Sliding Window Protocols

Part A: Go-Back-N Protocol

Q1)

B) In (FSM) diagram doesn't employ the modulo operator for sequence number updates. However, in our code, we've utilized the modulo operator for this purpose

C) The simulation runs without any errors for the case where Pc (probability of packet corruption) and Pl (probability of packet loss) are non-zero for both the DATA and ACK channels.

```
TIME: 13 Current window: [6, 7, 8, 9, 10] base = 6 nextseqnum = 11
--------------------
TIME: 14 RDT_SENDER: TIMEOUT OCCURED. Re-transmitting packets [6, 7, 8, 9, 10]
TIME: 14 DATA_CHANNEL : udt_send called for Packet(seq_num=6, payload=6, packet_length=100 bits, corrupted=False)
TIME: 14 DATA_CHANNEL : udt_send called for Packet(seq_num=7, payload=7, packet_length=100 bits, corrupted=False)
TIME: 14 DATA_CHANNEL : udt_send called for Packet(seq_num=8, payload=8, packet_length=100 bits, corrupted=False)
TIME: 14 DATA_CHANNEL : udt_send called for Packet(seq_num=9, payload=9, packet_length=100 bits, corrupted=False)
TIME: 14 DATA_CHANNEL : udt_send called for Packet(seq_num=10, payload=10, packet_length=100 bits, corrupted=False)
TIME: 14 TIMER STARTED for a timeout of  5
TIME: 14 RDT_SENDER: Got an ACK 5  for a packet in the old window. Ignoring it.
TIME: 14 SENDING APP: trying to send data 11
TIME: 14 RDT_SENDER: rdt_send() called for nextseqnum= 11  outside the current window. Refusing data.
TIME: 14 Current window: [6, 7, 8, 9, 10] base = 6 nextseqnum = 11
--------------------
TIME: 15 SENDING APP: trying to send data 11
TIME: 15 RDT_SENDER: rdt_send() called for nextseqnum= 11  outside the current window. Refusing data.
TIME: 15 Current window: [6, 7, 8, 9, 10] base = 6 nextseqnum = 11
--------------------
TIME: 16 RECEIVING APP: received data 6
TIME: 16 RDT_RECEIVER: got expected packet 6 . Sent ACK 6
TIME: 16 ACK_CHANNEL : udt_send called for Packet(seq_num=6, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 16 RECEIVING APP: received data 7
TIME: 16 RDT_RECEIVER: got expected packet 7 . Sent ACK 7
TIME: 16 ACK_CHANNEL : udt_send called for Packet(seq_num=7, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 16 RECEIVING APP: received data 8
TIME: 16 RDT_RECEIVER: got expected packet 8 . Sent ACK 8
TIME: 16 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 16 RECEIVING APP: received data 9
TIME: 16 RDT_RECEIVER: got expected packet 9 . Sent ACK 9
TIME: 16 ACK_CHANNEL : udt_send called for Packet(seq_num=9, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 16 RECEIVING APP: received data 10
TIME: 16 RDT_RECEIVER: got expected packet 10 . Sent ACK 10
TIME: 16 ACK_CHANNEL : udt_send called for Packet(seq_num=10, payload=ACK, packet_length=10 bits, corrupted=False)


Receiving application received 10 messages. Halting simulation.
================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 10
Total number of messages received by the Receiving App=10
Total number of DATA packets sent by rdt_Sender=15
Total number of re-transmitted DATA packets=5 (33.33% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=14
Total number of re-transmitted ACK packets=4 (28.57% of total packets sent)
Utilization for the DATA channel=9.38%
Utilization for the  ACK channel=0.87%
```

Q2)

For sub-part a, b and c

```
SIMULATION TIMES:
[4254, 4076, 4119, 4064, 4300]
Average Simulation Time:  4162.6


CHANNEL UTILIZATION:
[151.41043723554301, 150.9568204121688, 151.61446953143965, 152.75590551181102, 153.88372093023256]
Average channel utilization:  152.124270724239


FRACTION OF RETRANSMITTED PACKETS:
[84.38130725042696, 83.65025190963757, 83.87510008006404, 83.77899484536083, 84.78162309203566]
Average fraction of retransmitted packets:  84.09345543550502
```

A) Average time across 5 simulations is 4162.6

B) Average of data channel utilization across 5 simulations is 152.12427.

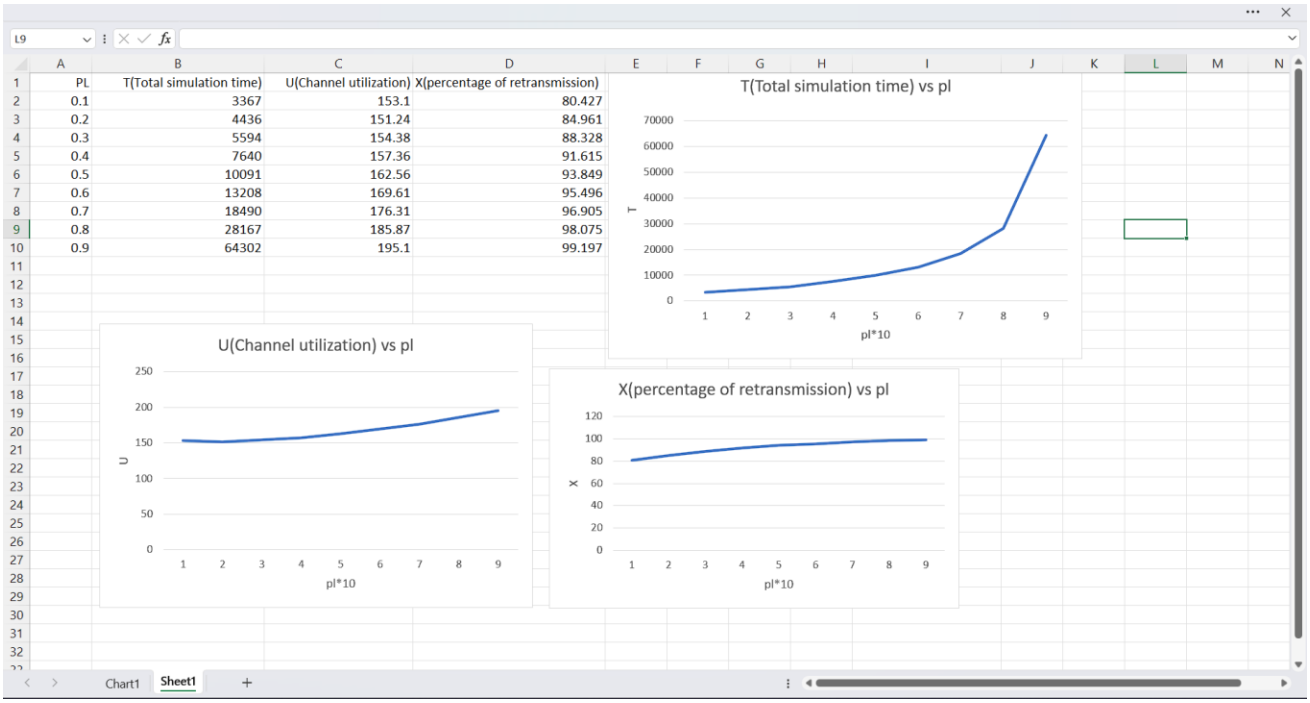
C) Average of fraction of packets that are simply retransmitted is 84.0934%.

Q3) For A, B and C graphs and data plotted below.

```
SIMULATION TIMES:
[3367, 4436, 5594, 7640, 10091, 13208, 18490, 28167, 64302]


CHANNEL UTILIZATION:
[153.1, 151.24, 154.38, 157.36, 162.56, 169.61, 176.31, 185.87, 195.1]

FRACTION OF RETRANSMITTED PACKETS:
[80.427, 84.961, 88.328, 91.615, 93.849, 95.496, 96.905, 98.075, 99.197]
PS D:\Academic\CS212\LAB5>
```



Q4)

A) Here, we can refer ‘best performance’ as optimal values of window size and k for which we can send all the packets in least amount of time so that data transfer can be faster

B) If window size is too small, then the data transfer can be slow because we need to send packet one by one to get ack it will be similar to RDT 3.0. And if the window size is to big equal to k, then when one packet is lost but others are received, then still we need to send all the packets again. This will multiply the time, therefore values cannot be much smaller and much bigger.

C)

```
SIMULATION TIMES:
[600, 465, 608, 599, 558, 760, 710, 620, 628, 437, 758, 464, 530, 559, 402]
SIMULATION TIMES:
[607, 632, 539, 561, 608, 678, 535, 631, 465, 536, 730, 482, 703, 496, 636]
SIMULATION TIMES:
[675, 494, 500, 434, 485, 783, 318, 445, 475, 489, 602, 401, 438, 407, 338]
SIMULATION TIMES:
[735, 443, 601, 557, 418, 709, 449, 425, 573, 527, 629, 712, 628, 540, 470]
SIMULATION TIMES:
[387, 502, 614, 376, 507, 743, 496, 448, 409, 539, 559, 582, 578, 577, 528]
SIMULATION TIMES:
[502, 363, 633, 421, 523, 544, 474, 560, 683, 526, 651, 681, 443, 597, 480]
N:  [3, 4, 5, 6, 7, 2, 3, 4, 5, 6, 2, 3, 4, 5, 6]
K:  [16, 16, 16, 16, 16, 24, 24, 24, 24, 24, 32, 32, 32, 32, 32]
```

600	465	608	599	558	760	710	620	628	437	758	464	530	559	402
607	632	539	561	608	678	535	631	465	536	730	482	703	496	636
675	494	500	434	485	783	318	445	475	489	602	401	438	407	338
735	443	601	557	418	709	449	425	573	527	629	712	628	540	470
387	502	614	376	507	743	496	448	409	539	559	582	578	577	528
502	363	633	421	523	544	474	560	683	526	651	681	443	597	480
584.333	**483.167**	**582.5**	**491.333**	**516.5**	**702.833**	**497**	**521.5**	**538.833**	**509**	**654.833**	**553.667**	**553.333**	**529.333**	**475.667**

As seen from above data tables best values of (N, K) I will choose is (N=6, K=32) following with (N=4, K=16) and (N=3, K=24).

As I run this simulation for many times with different N and K.I have stored the different simulation times and taking average of those values I came to conclusion.

Part B: Selective Repeat control

Q5) The Selective Repeat protocol, unlike the Go-Back-N protocol, is a sliding window protocol that allows the sender to send multiple packets without waiting for acknowledgment of each packet.

Sender State:

1. It sends a packet and sets a timer for this packet.

2. It keeps sending packets as long as the send window is not full.

3. If a timeout occurs for a packet, it resends the packet and restarts the timer.

4. When an ACK is received for a packet, it marks the packet as acknowledged and moves the window forward to include the next unacknowledged packet. If the ACK is lost or corrupted, the sender resends the packet.

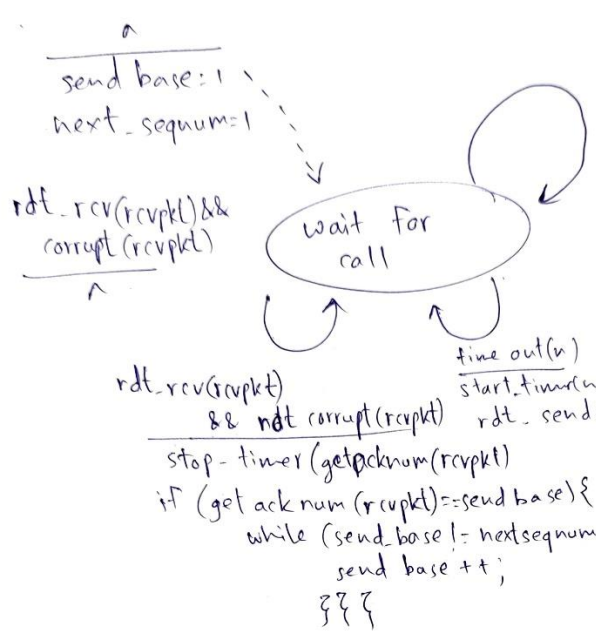
5. When all packets are acknowledged, the process stops.

Receiver State:

1. If the received packet is the one expected, it moves the receiver window forward and updates the expected packet number for the next received packet.

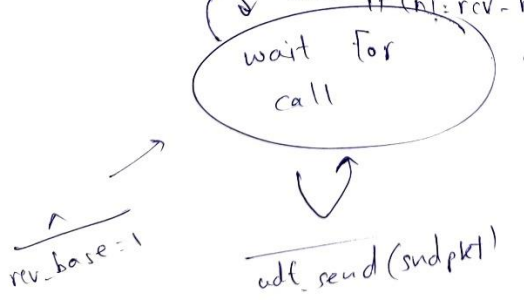
2. If the receiver window is full, it sends an ACK for the highest acknowledged packet based on the sequence number and stops receiving new packets.

Sender's Side




```
rdt_send(data)
if(next seqnum < base sen
  sndpkt[next seqnum]
  -make-pkt(nextseqnum,
      data, checksum
  udt_send(sndpkt[nextseqn
start_timer(next_seqnum)
  nextseqnum ++}
else {
  refuse data }
```

Reciver side




```
else {
  do {
    extract(rcvpkt,n,data)
    deliver_data(data)
      rcv_base ++}
  while (rcvbase != first
      pkt n)
else if (n < rcv_base){
  snd pkt = make pkt
    (n,ACK, checksum)
    udt_send(sndpkt)
      }
```

Q6)

A) PC = 0, PL=0

```
--------------------
TIME: 1001 TIMER STOPPED for packet. 5
TIME: 1001 RDT_RECEIVER: rdt() called for seq_num= 7  within current window. Sending ACK.
TIME: 1001 ACK_CHANNEL : udt_send called for Packet(seq_num=7, payload=ACK, packet_length=10 bits, corrupted=False)

TIME: 1001 RDT_RECEIVER: Currently buffered packets:
TIME: 1001 RDT_RECEIVER: Packet with seq_num= 7  and payload= 999


base 7
TIME: 1001 RECEIVING APP: received data 999
TIME: 1001 SENDING APP: trying to send data 1001
TIME: 1001 DATA_CHANNEL : udt_send called for Packet(seq_num=9, payload=1001, packet_length=100 bits, corrupted=False)
TIME: 1001 TIMER STARTED for a timeout of  5 for packet 9
TIME: 1001 Current window: [6, 7, 8, 9, 10] base = 6 nextseqnum = 10
TIME: 1002 TIMER STOPPED for packet. 6
TIME: 1002 RDT_RECEIVER: rdt() called for seq_num= 8  within current window. Sending ACK.
TIME: 1002 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)

TIME: 1002 RDT_RECEIVER: Currently buffered packets:
TIME: 1002 RDT_RECEIVER: Packet with seq_num= 8  and payload= 1000


base 8
TIME: 1002 RECEIVING APP: received data 1000


Receiving application received 1000 messages. Halting simulation.
================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 1001
Total number of messages received by the Receiving App=1000
Total number of DATA packets sent by rdt_Sender=1001
Total number of re-transmitted DATA packets=0 (0.00% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=1000
Total number of re-transmitted ACK packets=0 (0.00% of total packets sent)
Utilization for the DATA channel=9.99%
Utilization for the  ACK channel=1.00%
```

B) PC = 0.5, PL=0

```
TIME: 6866 Current window: [4, 5, 6, 7, 8] base = 4 nextseqnum = 9
--------------------
TIME: 6867 DATA_CHANNEL : udt_send called for Packet(seq_num=8, payload=1000, packet_length=100 bits, corrupted=False)
TIME: 6867 TIMER STARTED for a timeout of  5 for packet 8
TIME: 6867 SENDING APP: trying to send data 1001
TIME: 6867 Current window: [4, 5, 6, 7, 8] base = 4 nextseqnum = 9
--------------------
TIME: 6868 DATA_CHANNEL : udt_send called for Packet(seq_num=4, payload=996, packet_length=100 bits, corrupted=False)
TIME: 6868 TIMER STARTED for a timeout of  5 for packet 4
TIME: 6868 DATA_CHANNEL : Packet(seq_num=4, payload=****, packet_length=100 bits, corrupted=True) was corrupted!
TIME: 6868 RDT_RECEIVER: rdt_rcv() called for seq_num= 7  already delivered to app. Still Sending ACK.
TIME: 6868 ACK_CHANNEL : udt_send called for Packet(seq_num=7, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 6868 SENDING APP: trying to send data 1001
TIME: 6868 Current window: [4, 5, 6, 7, 8] base = 4 nextseqnum = 9
--------------------
TIME: 6869 DATA_CHANNEL : udt_send called for Packet(seq_num=5, payload=997, packet_length=100 bits, corrupted=False)
TIME: 6869 TIMER STARTED for a timeout of  5 for packet 5
TIME: 6869 RDT_RECEIVER: rdt() called for seq_num= 8  within current window. Sending ACK.
TIME: 6869 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)

TIME: 6869 RDT_RECEIVER: Currently buffered packets:
TIME: 6869 RDT_RECEIVER: Packet with seq_num= 8  and payload= 1000


base 8
TIME: 6869 RECEIVING APP: received data 1000


Receiving application received 1000 messages. Halting simulation.
================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 1000
Total number of messages received by the Receiving App=1000
Total number of DATA packets sent by rdt_Sender=7027
Total number of re-transmitted DATA packets=6027 (85.77% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=3518
Total number of re-transmitted ACK packets=0 (0.00% of total packets sent)
Utilization for the DATA channel=10.23%
Utilization for the  ACK channel=0.51%
PS D:\Academic\CS212\LAB5>
```

C) PC = 0.5, PL=0.5

```
TIME: 27268 DATA_CHANNEL : Packet(seq_num=7, payload=999, packet_length=100 bits, corrupted=False) was lost!
TIME: 27268 DATA_CHANNEL : udt_send called for Packet(seq_num=8, payload=1000, packet_length=100 bits, corrupted=False)
TIME: 27268 TIMER STARTED for a timeout of  5 for packet 8
TIME: 27268 RDT_RECEIVER: rdt_rcv() called for seq_num= 5  already delivered to app. Still Sending ACK.
TIME: 27268 ACK_CHANNEL : udt_send called for Packet(seq_num=5, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 27268 ACK_CHANNEL : Packet(seq_num=5, payload=****, packet_length=10 bits, corrupted=True) was corrupted!
TIME: 27268 ACK_CHANNEL : Packet(seq_num=5, payload=****, packet_length=10 bits, corrupted=True) was lost!
TIME: 27268 SENDING APP: trying to send data 1001
TIME: 27268 Current window: [4, 5, 6, 7, 8] base = 4 nextseqnum = 9
--------------------
TIME: 27269 SENDING APP: trying to send data 1001
TIME: 27269 Current window: [4, 5, 6, 7, 8] base = 4 nextseqnum = 9
--------------------
TIME: 27270 DATA_CHANNEL : udt_send called for Packet(seq_num=4, payload=996, packet_length=100 bits, corrupted=False)
TIME: 27270 TIMER STARTED for a timeout of  5 for packet 4
TIME: 27270 DATA_CHANNEL : Packet(seq_num=4, payload=****, packet_length=100 bits, corrupted=True) was corrupted!
TIME: 27270 RDT_RECEIVER: rdt() called for seq_num= 8  within current window. Sending ACK.
TIME: 27270 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)

TIME: 27270 RDT_RECEIVER: Currently buffered packets:
TIME: 27270 RDT_RECEIVER: Packet with seq_num= 8  and payload= 1000


base 8
TIME: 27270 RECEIVING APP: received data 1000


Receiving application received 1000 messages. Halting simulation.
================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 1000
Total number of messages received by the Receiving App=1000
Total number of DATA packets sent by rdt_Sender=27431
Total number of re-transmitted DATA packets=26431 (96.35% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=6764
Total number of re-transmitted ACK packets=0 (0.00% of total packets sent)
Utilization for the DATA channel=10.06%
Utilization for the  ACK channel=0.25%
PS D:\Academic\CS212\LAB5>
```

Q7)

Protocol_GBN: simulation time is 4433

```
--------------------
TIME: 4433 RECEIVING APP: received data 998
TIME: 4433 RDT_RECEIVER: got expected packet 6 . Sent ACK 6
TIME: 4433 ACK_CHANNEL : udt_send called for Packet(seq_num=6, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 4433 RECEIVING APP: received data 999
TIME: 4433 RDT_RECEIVER: got expected packet 7 . Sent ACK 7
TIME: 4433 ACK_CHANNEL : udt_send called for Packet(seq_num=7, payload=ACK, packet_length=10 bits, corrupted=False)
TIME: 4433 ACK_CHANNEL : Packet(seq_num=7, payload=****, packet_length=10 bits, corrupted=True) was corrupted!
TIME: 4433 RECEIVING APP: received data 1000
TIME: 4433 RDT_RECEIVER: got expected packet 8 . Sent ACK 8
TIME: 4433 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)


Receiving application received 1000 messages. Halting simulation.
P1:  0.2
Pc:  0.2


================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 1002
Total number of messages received by the Receiving App=1000
Total number of DATA packets sent by rdt_Sender=3762
Total number of re-transmitted DATA packets=2760 (73.37% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=3019
Total number of re-transmitted ACK packets=2019 (66.88% of total packets sent)
Utilization for the DATA channel=84.86%
Utilization for the  ACK channel=0.68%
```

Protocol_SR: simulation time is 4187

```
--------------------
TIME: 4187 RDT_RECEIVER: rdt() called for seq_num= 8  within current window. Sending ACK.
TIME: 4187 ACK_CHANNEL : udt_send called for Packet(seq_num=8, payload=ACK, packet_length=10 bits, corrupted=False)

TIME: 4187 RDT_RECEIVER: Currently buffered packets:
TIME: 4187 RDT_RECEIVER: Packet with seq_num= 8  and payload= 1000


base 8
TIME: 4187 RECEIVING APP: received data 1000


Receiving application received 1000 messages. Halting simulation.
Pl:  0.2
Pc:  0.2


================================================
 SIMULATION RESULTS:
================================================
Total number of messages sent by the Sending App= 1001
Total number of messages received by the Receiving App=1000
Total number of DATA packets sent by rdt_Sender=4341
Total number of re-transmitted DATA packets=3340 (76.94% of total packets sent)
Total number of ACK packets sent by rdt_Receiver=2789
Total number of re-transmitted ACK packets=0 (0.00% of total packets sent)
Utilization for the DATA channel=103.68%
Utilization for the  ACK channel=0.67%
```

Here total simulation time of GBN is greater than SR. It tells us that the SR protocol is better compared to that of the GBN protocol. Because GBN protocol works on the basis of cumulative approach.

Q8) When N=K,

```
TIME: 25 RDT_RECEIVER: Packet with seq_num= 9  and payload= 9
TIME: 25 RDT_RECEIVER: Packet with seq_num= 3  and payload= 19
TIME: 25 RDT_RECEIVER: Packet with seq_num= 5  and payload= 21
TIME: 25 RDT_RECEIVER: Packet with seq_num= 7  and payload= 7
TIME: 25 RDT_RECEIVER: Packet with seq_num= 12  and payload= 12
TIME: 25 RDT_RECEIVER: Packet with seq_num= 1  and payload= 17


TIME: 25 ACK_CHANNEL : Packet(seq_num=8, payload=****, packet_length=10 bits, corrupted=True) was corrupted!
TIME: 25 SENDING APP: trying to send data 25
TIME: 25 DATA_CHANNEL : udt_send called for Packet(seq_num=9, payload=25, packet_length=1000 bits, corrupted=Fals
TIME: 25 TIMER STARTED for a timeout of  5 for packet 9
Traceback (most recent call last):
  File "d:\Academic\CS212\LAB5\Protocol_SR.py", line 125, in timer_behavior
    self.timeout_action(seq_num)
  File "d:\Academic\CS212\LAB5\Protocol_SR.py", line 154, in timeout_action
    self.channel.udt_send(self.sndpkt[seq_num])
                          ~~~~~~~~~~~~^^^^^^^^^
KeyError: 5

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "d:\Academic\CS212\LAB5\Testbench.py", line 56, in <module>
    env.step()
  File "C:\Users\hp\AppData\Local\Programs\Python\Python311\Lib\site-packages\simpy\core.py", line 204, in step
    raise exc
KeyError: 5
```

The problem is when we take N>K/2, is all the packets from sequence numbers 0 to K gets competed then again, the sequence numbers will start from 0, so there is high chance that the sender and receiver may confuse the packet from new sequence range and packets gets mixed up. Therefore, we take value of N<=K/2.