National University of Computer and Emerging Sciences (Lahore Campus)

Quiz 3: Transport Layer (Chapter 3)

Name: ______________ **Roll No:** ______________ **Section:** BSE-6B1 (Spring 2026)

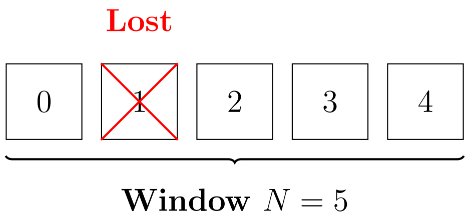
1. (5 points) **Go-Back-N vs Selective Repeat Analysis**

   Consider a pipelined protocol over a link with a very high Bandwidth-Delay Product.

   - Sender sends packets $0, 1, 2, 3, 4$. Packet 1 is lost. All other packets and ACKs arrive correctly. Timeout occurs after Packet 4 is sent. Window Size $N = 5$. Sequence Numbers: $0, 1, 2, \ldots$

   (a) If using **Go-Back-N**, mathematically calculate the total number of packet transmissions (initial + retransmissions) required to successfully deliver packets 0 through 4. If using **Selective Repeat**, calculate the same.

   (b) Will **Selective Repeat** operate correctly if N = 5 and k = 3? Consider the SR window size constraint, create a scenerio where it fails and explain it.



Solution:

(a)

1. **Go-Back-N:** 1. Sender sends 0, 1, 2, 3, 4. (5 transmissions). 2. Receiver gets 0 (ACKs 0). Receiver expects 1, gets 2, 3, 4 (discards them, re-ACKs 0). 3. Sender Timeout for Packet 1. 4. GBN retransmits from base: Sends 1, 2, 3, 4. (4 transmissions). Total transmissions $= 5 + 4 = 9$.
2. **Selective Repeat:** 1. Sender sends 0, 1, 2, 3, 4. (5 transmissions). 2. Receiver gets 0 (ACKs 0). Receiver misses 1. Receiver gets 2, 3, 4 (Buffers them, ACKs 2, 3, 4 selectively). 3. Sender Timeout for Packet 1. 4. Sender retransmits ONLY 1. (1 transmission). Total transmissions $= 5 + 1 = 6$.

(b)

SR Protocol Constraint: Window Size N must satisfy $N \leq \frac{1}{2} \times 2^k$.
Here $2^k = 8$. Max N should be 4.
Given $N = 5$. Problem:

1. Sender sends 0,1,2,3,4.
2. Rx receives all, sends ACKs, moves window to [5,6,7,0,1].
3. All ACKs lost.

4. Sender times out, resends Packet 0.
5. Receiver sees Packet 0. Is this a retransmission of old 0 or new 0 (in window [5,6,7,0,1])? It cannot distinguish. Protocol fails.

2. (10 points) **TCP Sequence Numbers & Re-ordering**

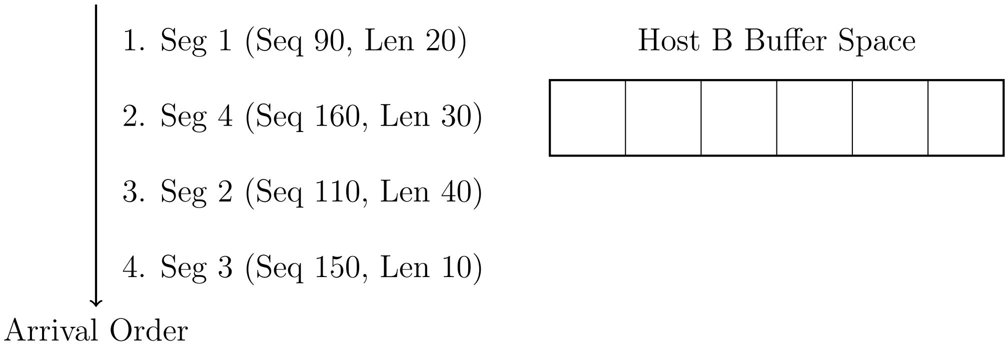
Host A sends 4 segments to Host B over a TCP connection.

- Segment 1: Seq = 90, Length = 20 bytes.
- Segment 2: Seq = 110, Length = 40 bytes.
- Segment 3: Seq = 150, Length = 10 bytes.
- Segment 4: Seq = 160, Length = 30 bytes.

Assume the initial Sequence Number is 90. Host B receives them in this order: **Segment 1, Segment 4, Segment 2, Segment 3**.

(a) List the ACK number Host B sends immediately after receiving EACH segment in that specific order.

(b) Draw the receive buffer state after Segment 4 arrives but before Segment 2 arrives.



1. Seg 1 (Seq 90, Len 20)
2. Seg 4 (Seq 160, Len 30)
3. Seg 2 (Seq 110, Len 40)
4. Seg 3 (Seq 150, Len 10)

Arrival Order

Host B Buffer Space

Solution:

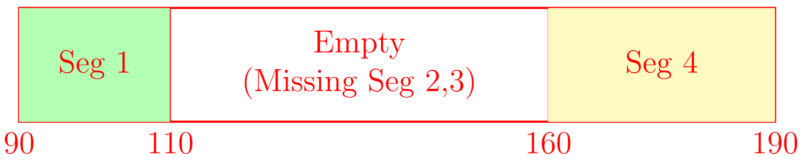
(a) ACK Generation:

1. Rx Seg 1 (90-109): Data is in order. Next expected is 110. **ACK = 110**.
2. Rx Seg 4 (160-189): Gap detected (missing 110-159). Buffer Seg 4. Send duplicate ACK. **ACK = 110**.
3. Rx Seg 2 (110-149): Gap partially filled. Still missing 150. Buffer Seg 2. Send duplicate ACK. **ACK = 150**.
4. Rx Seg 3 (150-159): Gap filled. We have 90 through 189 continuous bytes. Next expected is 190. **ACK = 190**.

(b) Receive Buffer State (After Seg 4, Before Seg 2):

State Description: Segment 1 is delivered/acknowledged. Segment 4 is buffered (out of order). There is a ”Hole” where Segment 2 and 3 should be.

Host B Buffer Content (Seq 90 to 190)


Seg 1
Empty
(Missing Seg 2,3)
Seg 4
90
110
160
190